Date: August 19, 2010

User's Manual
for the
CV-5000 .NET Library

Prepared by:

CONQUEST CONSULTING LLC

www.conquestsolutions.com

Copyright © 2010

This document contains information which is the property of Conquest Consulting LLC.
Except for rights expressly granted by contract, this document may not, in whole or in part, be duplicated or disclosed without prior written permission.

Revision History

Revision	Release Status		Date
1.0.0.1	Initial Release		2009-08-05
1.0.0.2	4.2	Added Activation Tool and Activation Log to the program menu. Removed auto creation of distribution license.	2009-11-18
	4.3	Added entire section which shows how to activate the library. Removed auto creation of distribution license.	
	4.4	Added entire section which explains trial mode.	
	5.3	Added “Stretch Image” size mode setting to the Picture Box	
	5.6.1	Modified the code to update the picture box.	
	6.1.2	Password no longer required.	
	6.1.6 & 6.1.7	Updated DebugMode_LogToFile to include the directories in which the debug files are stored.	
	6.1.8	Added section to explain the “Binary Dump” error logging.	
	6.2.5	Updated the bitmap property definition to include information on disposing of the bitmap.	
	6.4.63	Added “WriteImageToDisk” method.	
	A.3	Updated code to acquire the bitmap image.	
	5.4.2	Removed setting of password from sample code.	
	A.1, A.2, A.3	Removed setting of password from sample code.	
1.0.0.3	4.5	Added Deployment Section	2009-11-25
	4.5.1	Added information on installing activation tool on client computer.	
	5.4.2	Add note for user to copy CQActivation.dll to application directory.	
1.0.0.4	4.1	Run setup.exe instead of CV5000LibSetup.msi.	2010-08-19
	4.3	Updated activation section so that development/distribution option is no longer shown.	
	4.5.2	Added section on using the CV-5000 Library with 64 bit operating Systems	
	6.2.5	Added to TriggeredImageCameraNumber, TriggeredImageTotalStatus, and TriggeredImageMeasurementCount properties to DataEventArgs.	

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

TABLE OF CONTENTS

SECTION	TITLE	PAGE
1	End-User License Agreement for Conquest Consulting CV-5000 Series .NET Library	1
2	Preface	4
3	Overview of the CV-5000 .NET Library	5
3.1	Minimum System Requirements	5
4	Getting Started	6
4.1	Installing the CV-5000 .NET Library	6
4.2	Program Menu	8
4.3	Activating the CV-5000 .Net Library	8
4.4	Trial Mode	9
4.5	Deploying Your Application	9
4.5.1	Activation of CV-5000 .Net Library on Remote Computer	9
4.5.2	Using the Keyence CV-5000 with 64-bit Operating Systems	10
5	Your First CV-5000 Program	11
5.1	Setting up your Program	12
5.2	Adding a Reference to the CV-5000 .NET Library	13
5.3	Add Required GUI Components to the Application's Main Form	14
5.4	Connecting to the CV-5000 Device	15
5.4.1	Add CV5000_Series Namespace & Create an CV-5000 Library Object	15
5.4.2	Connecting to the CV	16
5.4.3	Creating Event Handlers	17
5.4.4	Updating the Status Textbox in a Thread-safe Manner	18
5.4.5	Closing Connections	19
5.4.6	Running the Application with Connection Events	19
5.5	Retrieving Measurement Data from the CV-5000 Device	20
5.5.1	Updating the Measurement ListBox in a ThreadSafe way	20
5.5.2	Add Data Received Event Handler	21
5.5.3	Setting up the Timer	22
5.5.4	Setting up the Start and Stop Buttons	22
5.5.5	Running the Application with the Trigger Data	23
5.6	Retrieving Image Data from the CV-5000 Device	24
5.6.1	Updating the PictureBox in a Thread-safe Manner	24
5.6.2	Add Image Data to the Data Received Handler	25
5.6.3	Running the Complete Sample Program	26
6	The CV-5000 .NET Library Reference	27
6.1	Properties	27

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.1.1 Property: Name 27
6.1.2 Property: Password 27
6.1.3 Property: RemoteHost 28
6.1.4 Property: RemotePort 28
6.1.5 Property: RemoteImagePort 28
6.1.6 Property: DebugMode_LogToFile 29
6.1.7 Property: ErrorLogging 29
6.1.8 Property: BinaryDump 29
6.2 Events 30
6.2.1 Event: ErrorOccurred 30
6.2.2 Event: ConnectionEstablished 32
6.2.3 Event: ImageConnectionEstablished 32
6.2.4 Event: DataSent 32
6.2.5 Event: DataReceived 33
6.2.6 Event: ConnectionLost 35
6.2.7 Event: ConnectionClosedByProgram 35
6.3 Communication Methods 36
6.3.1 Method: Connect 36
6.3.2 Method: CloseConnection 36
6.3.3 Method: SendRawCommand 37
6.3.4 Method: SendRawCommandArray 37
6.4 CV Commands 38
6.4.1 Method: Trigger 40
6.4.2 Method: ChangeModeToRun 42
6.4.3 Method: ChangeModeToProgram 43
6.4.4 Method: RepeatOutputMeasurementResult 44
6.4.5 Method: SwitchProgramNumber 46
6.4.6 Method: ReadCurrentProgramNumber 47
6.4.7 Method: SwitchWindowNumber 48
6.4.8 Method: ReadCurrentWindowNumber 49
6.4.9 Method: SwitchDisplayTemplate 50
6.4.10 Method: SwitchResultDisplay 51
6.4.11 Method: SwitchPageNumber 52
6.4.12 Method: SwitchPageNext 53
6.4.13 Method: SwitchPagePrevious 54
6.4.14 Method: SwitchScreen 55
6.4.15 Method: SaveScreenCaptureImage 56
6.4.16 Method: RequestScreenCapture 57
6.4.17 Method: RegisterImage 58
6.4.18 Method: RetrieveListRegisteredImagesCurrentProgram 59
6.4.19 Method: RetrieveListRegisteredImages 60
6.4.20 Method: RequestImageData 61
6.4.21 Method: RequestImageToSDCard 62
6.4.22 Method: ReadAllImageDataInBuffer 63

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.23 Method: WriteProgramDataToSDDevice 64
6.4.24 Method: WriteProgramDataToCurrentProgram 65
6.4.25 Method: WriteRegisteredImageToSDDevice 66
6.4.26 Method: WriteRegisteredImageToCurrentProgram 67
6.4.27 Method: WriteLibraryDataToSDDevice 68
6.4.28 Method: WriteLibraryDataToCurrentProgram 69
6.4.29 Method: ReadProgramDataFromSDDevice 70
6.4.30 Method: ReadProgramDataFromCurrentProgram 71
6.4.31 Method: ReadRegisteredImageFromSDDevice 72
6.4.32 Method: ReadRegisteredImageFromCurrentProgram 73
6.4.33 Method: ReadLibraryDataFromSDDevice (TBD – Update DataEventArgs) 74
6.4.34 Method: ReadLibraryDataFromCurrentProgram (TBD – Update DataEventArgs) 75
6.4.35 Method: RetrieveListOfLibraryNumbers 76
6.4.36 Method: WriteGlobalSettingsData 77
6.4.37 Method: ReadGlobalSettingsData 78
6.4.38 Method: WriteBinaryFilterData 79
6.4.39 Method: ReadBinaryFilterData 80
6.4.40 Method: WriteWindowCalculationValue 81
6.4.41 Method: ReadWindowCalculationValue 82
6.4.42 Method: WriteToCommandMemory 83
6.4.43 Method: ReadFromCommandMemory 84
6.4.44 Method: SetInitialValuesForCommandMemory 85
6.4.45 Method: ChangeShutterSpeed 86
6.4.46 Method: ChangeCameraSensitivity 87
6.4.47 Method: SetTriggerDelayValue 88
6.4.48 Method: ChangeExtensionUnitLightIntensity 89
6.4.49 Method: SetExtensionPatternCounter 90
6.4.50 Method: SetDateTime 91
6.4.51 Method: ReadDateTime 92
6.4.52 Method: RecalculateReferenceValues 93
6.4.53 Method: SaveProgramSettings 94
6.4.54 Method: Reset 95
6.4.55 Method: TriggerInputEnable 96
6.4.56 Method: TriggerInputDisable 97
6.4.57 Method: EnableCommunicationsPort 98
6.4.58 Method: DisableCommunicationsPort 99
6.4.59 Method: ChangePassword 100
6.4.60 Method: RemoteControlPseudoInput 101
6.4.61 Method: EchoText 102
6.4.62 Method: SaveStatisticalData 103
6.4.63 Method: WriteImageToDisk 103
6.5 Support 104

APPENDIX A - Code Listings 105

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

A.1 Connection Code Listing.....106
A.2 Measurement Data Code Listing.....108
A.3 Complete Code Listing.....110

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

1 END-USER LICENSE AGREEMENT FOR CONQUEST CONSULTING CV-5000 SERIES .NET LIBRARY

IMPORTANT-READ CAREFULLY: This Software End-User License Agreement ("EULA") is a legal agreement between you (either an individual person or a single legal entity, hereinafter referred to as "CUSTOMER") and CONQUEST CONSULTING, LLC (hereinafter referred to as "CONQUEST") with offices at 30 Tower Lane Avon, Connecticut, 06001 for the .NET software product that accompanies this EULA, including any associated media, printed materials, electronic documentation and/or derivative works (including materials created with a so called level editor, if included) collectively referred to as the "Software Product". As determined by CONQUEST and to the extent provided by any accompanying maintenance agreement, the Software Product includes any software updates, add-on components, web services and/or supplements that CONQUEST may provide or make available to CUSTOMER after the date CUSTOMER obtains their initial copy of the Software Product to the extent that such items are not accompanied by a separate license agreement or terms of use. By obtaining, installing, downloading, accessing or otherwise using the Software Product, CUSTOMER agrees to be bound by the terms of this EULA. If CUSTOMER does not agree to the terms of this EULA, do not install, access or use the Software Product and promptly return the unused Software Product to the place of purchase or contact CONQUEST Customer Service for a full refund of the purchase price within 30 days of the original purchase.

Be advised that the Software Product is the copyrighted work of CONQUEST. All rights reserved, except as expressly stated herein. The Software Product is solely for use by an end user according to the terms of this EULA. Any use, reproduction or redistribution of the Software Product not in accordance with the terms of the License Agreement is expressly prohibited.

1. SOFTWARE PRODUCT LICENSE

The Software Product is protected by domestic and international intellectual property laws and treaties. The Software Product is licensed, not sold. As such, this EULA confers no title or ownership in the Software Product. Pursuant to CUSTOMER accepting all the terms and conditions of this EULA, CONQUEST grants to CUSTOMER a personal, exclusive, license to install and use the Software Product on any computer owned by CUSTOMER and operated by CUSTOMER and its employs.

Without the expressed written consent of CONQUEST, CUSTOMER is not entitled to sell, grant a security interest in or transfer reproductions of the Program to other parties (either an individual person or a single legal entity) not owned and operated by CUSTOMER, in any way, nor to rent, lease or license the Program to others (either an individual person or a single legal entity) not owned and operated by CUSTOMER.

If CUSTOMER has acquired this Software Product in a CONQUEST License Pack or Corporate Site License Agreement, CUSTOMER may make the number of additional copies of the computer software portion of the Software Product as defined on that License Pack or Corporate Site License, and CUSTOMER may use each copy in the manner specified above and below.

2. OWNERSHIP

Unless otherwise noted, all right, title and interests in the Software Product are owned by CONQUEST, including any accompanying downloaded or printed materials. All intellectual property rights, including but not limited to patentable inventions, copyrights, trade secrets and trademarks are owned by CONQUEST. Nothing in this EULA gives CUSTOMER any ownership interests in the Software Product. CUSTOMER may make one copy of the Software Product for back-up and archival purposes. CUSTOMER may make one copy the printed materials accompanying the Software Product.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

CUSTOMER may not remove, modify or alter any CONQUEST copyright or trademark notices from any part of the Software Product, including but not limited to any such notices contained in the physical and/or electronic media or documentation, in the CONQUEST Setup Wizard dialogue or 'about' boxes, in any of the runtime resources and/or in any web-presence or web-enabled notices, code or other embodiments originally contained in or otherwise created by the Software Product.

The Software Product may contain certain materials licensed to CONQUEST and these licensors may individually and separately protect their rights in the event of any violation of this Agreement. By agreeing to the terms of this EULA CUSTOMER agrees to the terms of the license of such materials licensed to CONQUEST.

3. CONFIDENTIALITY

CUSTOMER acknowledges that the Software Product includes confidential and proprietary information of CONQUEST. CUSTOMER will take all reasonable precautions necessary to safeguard the confidentiality of the Software Product. CUSTOMER will not modify, decompile, disassemble, or reverse engineer the Software Product, or assist any other party in doing the same. Additionally, CUSTOMER and its employs will not use the Software Product to create a new product, modify an existing product or to assist any other party in doing same to compete, either directly or indirectly, with any products and/or services of CONQUEST.

CUSTOMER acknowledges that any unauthorized use or disclosure of the Software Product will cause irreparable harm to CONQUEST. If an unauthorized use or disclosure occurs, CUSTOMER will promptly inform CONQUEST and take all steps that are necessary to recover the Software Product and to prevent its subsequent unauthorized use or dissemination. CONQUEST shall be entitled to equitable relief, including reasonable attorney's fees and expenses, to enforce this requirement.

4. UPGRADES AND SUBSCRIPTION

If the Software Product is labeled or otherwise identified by CONQUEST as an "upgrade" or "subscription," CUSTOMER must be properly licensed to use a product identified by CONQUEST as being eligible for the upgrade in order to use the Software Product. A Software Product, labeled or otherwise identified by CONQUEST as an upgrade, replaces and/or supplements the product that formed the basis for CUSTOMERs eligibility for such upgrade. CUSTOMER may use the resulting upgraded product only in accordance with the terms of this EULA. If the Software Product is an upgrade of a component of a package of software programs that CUSTOMER licensed as a single product, the Software Product may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

5. TERMINATION

Without prejudice to any of CONQUEST'S other rights, CONQUEST may immediately terminate this EULA if CUSTOMER fails to comply with the terms and conditions of this EULA. In such event, CUSTOMER must immediately cease all use of the Software Product and delete any and all copies of the Software Product and all of its component parts.

6. MISCELLANEOUS

THE SOFTWARE PRODUCT IS PROVIDED "AS IS" AND CONQUEST CONSULTING DISCLAIMS ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. CONQUEST MAKES NO WARRANTY THAT ALL ERRORS HAVE BEEN OR CAN BE ELIMINATED FROM THE SOFTWARE PRODUCT OR DOCUMENTATION. CUSTOMER ACKNOWLEDGES AND AGREES THAT ANY USE OF THE SOFTWARE PRODUCT IS AT THE SOLE RISK OF CUSTOMER.

IN NO EVENT SHALL CONQUEST BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

OF OR INABILITY TO USE THE SOFTWARE PRODUCT OR THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF CONQUEST HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, CONQUEST'S ENTIRE LIABILITY UNDER ANY PROVISION OF THIS EULA SHALL BE LIMITED TO THE AMOUNT CUSTOMER ACTUALLY PAID TO CONQUEST FOR THE SOFTWARE PRODUCT OR SERVICE THAT DIRECTLY CAUSED THE DAMAGE.

CUSTOMER and CONQUEST are independent parties. Nothing in this EULA shall be construed as making CUSTOMER an employee, agent or legal representative of CONQUEST.

This EULA constitutes the complete and entire statement of all terms, conditions and representations of the agreement between CUSTOMER and CONQUEST with respect to the Software Product. CUSTOMER agrees to abide by all applicable laws with regard to the use, sale and export of the Software Product.

This EULA is governed by the laws of the State of Connecticut without regard to conflicts of law principles. CUSTOMER hereby consents to the exclusive jurisdiction of the courts of the State of Connecticut for the resolution of any dispute arising out of or related to this EULA and/or use of the Product Software.

The failure of CONQUEST to insist, in any one or more instances, upon the performance of any of the terms of this EULA or to exercise any right hereunder, shall not be construed as a waiver of the future performance of any such term or the future exercise of such right.

In the event that any provision of this EULA shall be held by a court or other tribunal of competent jurisdiction to be unenforceable (in whole or in part), such provision will be enforced to the maximum extent permissible and the remaining portions of this EULA shall be construed in all respects as if such invalid or unenforceable provisions were omitted and shall remain in full force and effect.

The Software Product may not be re-exported, downloaded or otherwise exported into (or to a national or resident of) any country to which the U.S. has embargoed goods, or to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Denial Orders. By agreeing to the terms of this EULA, CUSTOMER is agreeing to the foregoing and CUSTOMER is representing and warranting that CUSTOMER is not located in, under the control of, or a national or resident of any such country or on any such list.

CUSTOMER AGREES THAT CUSTOMER HAS READ AND AGREES TO BE BOUND BY ALL TERMS AND CONDITIONS OF THIS EULA.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

2 PREFACE

The CV-5000 Series .NET Library Developer's Guide is not intended to replace the CV-5202 User's Manual. This guide assumes the user is familiar with the basic setup and operation of the CV-5000 Series Camera System.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

3 OVERVIEW OF THE CV-5000 .NET LIBRARY

The CV-5000 .NET Library allows .NET programmers to quickly and easily communicate with the Keyence CV-5000 via Ethernet. The CV-5000 .NET Library contains a single control named CV5000_Series.dll that encapsulates all of the Ethernet-based functionality of the Keyence CV-5000 series cameras.

The CV-5000 .NET Library supports a vast array of features including, but not limited to:

- Managing the TCP connection to your CV-5000 series device
- Support for all CV-5000 Ethernet-based functions in RUN and PROGRAM modes
- Data and image acquisition methods

The CV-5000 .NET Library provides support for a range of methods from sending low-level raw commands to execution of high-level CV-5000 functionality.

The CV-5000 .NET Library allows your application maximum CPU time when waiting for processes that take an extended period of time (e.g. screen captures). Communication methods return immediately after data is sent to the CV-5000 device and whenever the data is available from the CV-5000 .NET Library, an event is triggered to allow your application to access the received data.

3.1 Minimum System Requirements

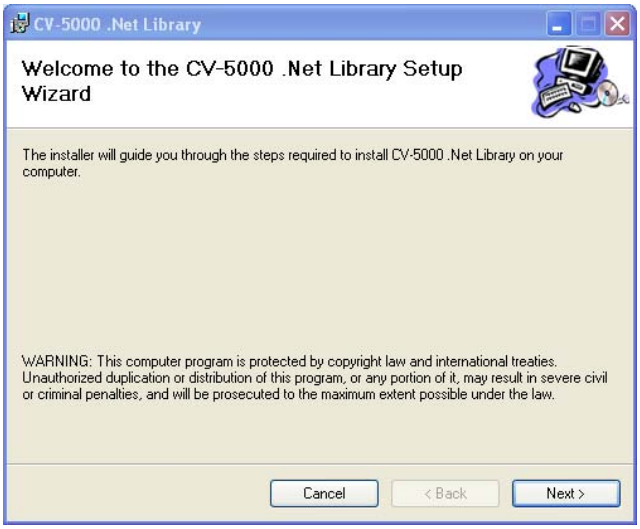
- Microsoft Windows XP or Windows Vista
- Microsoft .NET Framework 2.0
- Ethernet interface

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

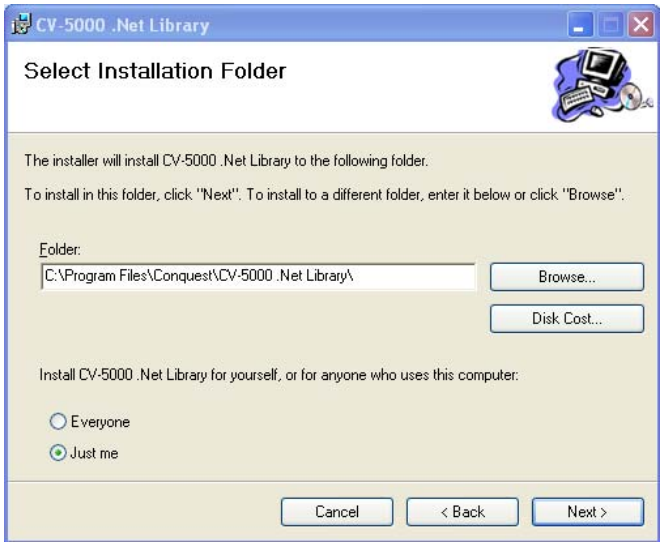
4 GETTING STARTED

4.1 Installing the CV-5000 .NET Library

Locate the Installation Program (setup.exe) that you downloaded from the website. (Make sure that CV5000LibSetup.msi is in the same directory as setup.exe.) Run the installation program and follow the commands below:

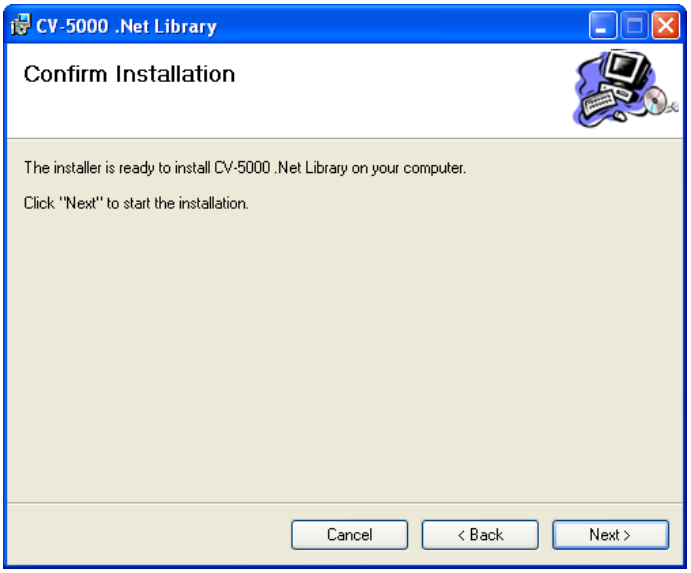

CV-5000 .Net Library
Welcome to the CV-5000 .Net Library Setup Wizard
The installer will guide you through the steps required to install CV-5000 .Net Library on your computer.
WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.
Cancel
< Back
Next >

Click **Next** to continue.

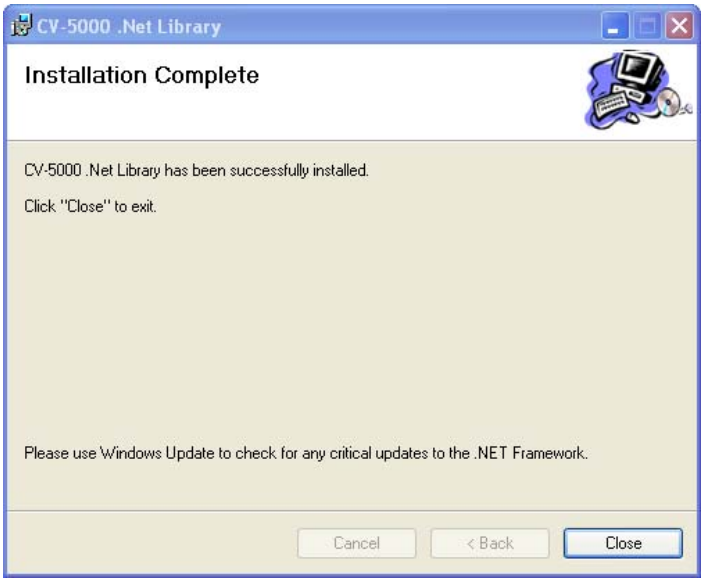

CV-5000 .Net Library
Select Installation Folder
The installer will install CV-5000 .Net Library to the following folder.
To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".
Folder:
C:\Program Files\Conquest\CV-5000 .Net Library\
Browse...
Disk Cost...
Install CV-5000 .Net Library for yourself, or for anyone who uses this computer:
Everyone
Just me
Cancel
< Back
Next >

Select the installation folder or leave the default that was provided and click **Next** to continue.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.


CV-5000 .Net Library
Confirm Installation
The installer is ready to install CV-5000 .Net Library on your computer.
Click "Next" to start the installation.
Cancel
< Back
Next >

Confirm the installation by clicking **Next** as shown below


CV-5000 .Net Library
Installation Complete
CV-5000 .Net Library has been successfully installed.
Click "Close" to exit.
Please use Windows Update to check for any critical updates to the .NET Framework.
Cancel
< Back
Close

Once the installation has completed, click **Close**.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

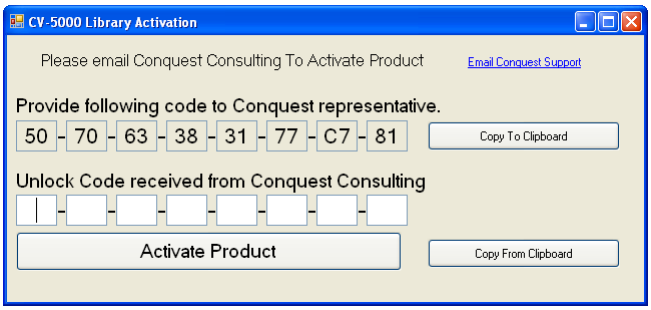
4.2 Program Menu

The following shortcut will be provided in the Windows Program Menu as shown below:

- CV-5000 – This shortcut will open the directory where the CV-5000 .NET library is located. The .NET Library (CV5000_Series.dll) is a .NET assembly.
- CV5000_ActivationTool – This shortcut will launch an application that will activate the product.
- Activation Log – This shortcut will open a directory that will contain diagnostic information for any activation issues.

4.3 Activating the CV-5000 .Net Library

1. Select the CV5000_ActivationTool shortcut from the CV-5000 program menu and the following window should appear.


CV-5000 Library Activation
Please email Conquest Consulting To Activate Product
Email Conquest Support
Provide following code to Conquest representative.
50 - 70 - 63 - 38 - 31 - 77 - C7 - 81
Copy To Clipboard
Unlock Code received from Conquest Consulting
Activate Product
Copy From Clipboard

2. Click the “Copy To” Clipboard button to copy the code that appears on the screen.
3. Click the “Email Conquest Support” link on the top right corner of the form and the default email client should open with an email addressed to support@conquestsolutions.com.
4. Right-click in the email body and select “Paste”. A code like the following will appear in the email body: 4B-EF-57-F0-68-FF-F1-0F
5. Put your contact information in the email and send the email.
6. Conquest will respond to the email with an activation code. When the activation code arrives, copy the code to the windows clipboard and click on the “Copy from Clipboard” button to paste the activation code into the form.
7. Select either the Development Activation or Distribution Activation radio button. You must make the same selection that you made above in step 2.
8. Click the “Activate Product” button. If the activation is successful, a development license will be written to the computer.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

4.4 Trial Mode

Until the CV-5000 .Net library is activated, it will run in trial mode for 30 days. Programs run under trial mode will experience a dialog box every thirty(30) minutes prompting the user to enter the trial password obtained from Conquest Consulting.

4.5 Deploying Your Application

You must include the following files when deploying your application:

CV5000_Series.dll: This file contains all the properties, methods, and events for the CV-5000 .Net library. This file must reside in the same directory as the final application.

CQActivation.dll: This file handles the licensing for the CV-5000 .Net library. This file must reside in the same directory as the final application.

CV5000_Series.pdb: This file contains debugging information that can be used with the CV5000_Series.dll. This file is optional and if distributed it must reside in the same directory as the final application.

These files can be located in the CV-5000 .NET Library installation directory. Locate this directory by clicking on the CV5000 shortcut..

4.5.1 Activation of CV-5000 .Net Library on Remote Computer

The ActivationTool.exe application must be used on each computer that runs the CV-5000 .Net Library to activate the distribution license for that computer. There are three ways to install this activation tool on each computer:

1. Run the CV-5000 .Net Library Installation as described in section 4.1 on each computer that will run the application that utilizes the CV-5000 .Net Library. Then follow the instructions in section 4.3 of this manual for activating the CV-5000 .Net library.

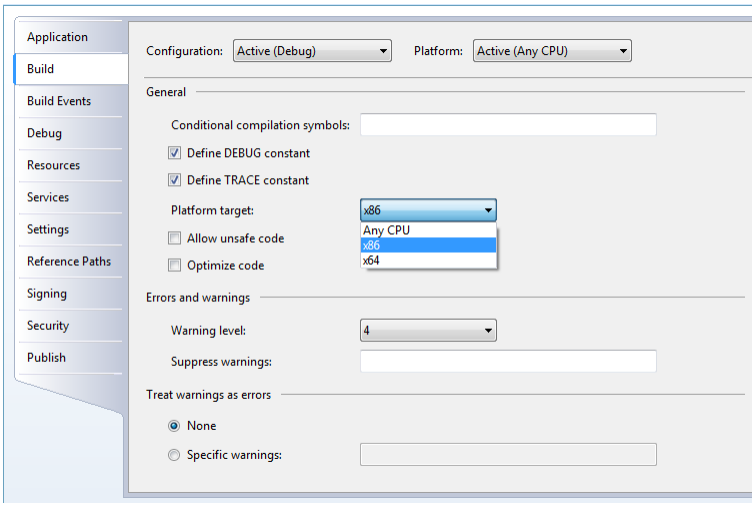
2. Manually copy and paste ActivationTool.exe along with CQActivation.dll to each computer that will run the application that utilizes the CV-5000 .Net Library. Then follow the instructions in section 4.3 of this manual for activating the CV-5000 .Net library.

3. Create a setup program that will distribute the ActivationTool.exe along with the application that utilizes the CV-5000 .Net library. Make sure the setup program places a copy of CQActivation.dll in the same directory as ActivationTool.exe. Then follow the instructions in section 4.3 of this manual for activating the CV-5000 .Net library on each computer where this setup program is run.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

4.5.2 Using the Keyence CV-5000 with 64-bit Operating Systems

If a .Net application that uses the CV-5000 library is run on a 64-bit version of Windows then the .Net application must be configured to always run in 32-bit mode. To do this, right-click on the project in visual studio and click on the build tab on the left. The following screen will be shown. Change the platform target to x86.



Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

5 YOUR FIRST CV-5000 PROGRAM

This program will demonstrate the use of the CV-5000 .NET Library in a simple program to log measurement data and part images sent from the CV-5000. This example assumes you are familiar with VB.NET and Visual Studio 2008.

NOTE - Before you create the sample program, you must set the IP address in the CV-5000 and configure the CV-5000 to output all measurement data and acquired images to be sent via the Ethernet.

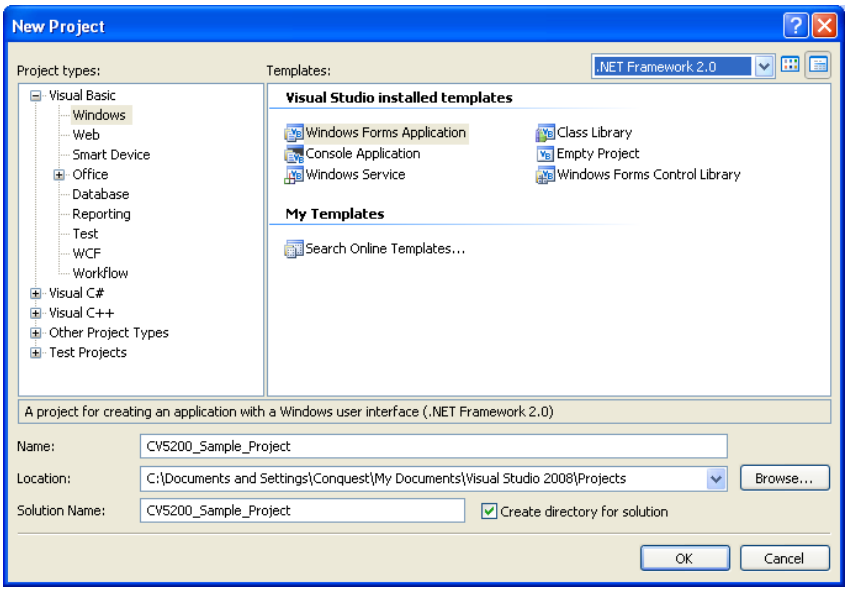
Configuring the CV-5000 is beyond the scope of this document and is not discussed here. Please consult the assistance of your companies IT department and refer to the CV-5000 User's Manual for more information.

The sample program will be responsible for triggering the CV-5000 every 5 seconds. When measurement and image data is received by the CV-5000 via the Ethernet connection it will be displayed to the user and saved to disk. This program will use a single camera and a single window.

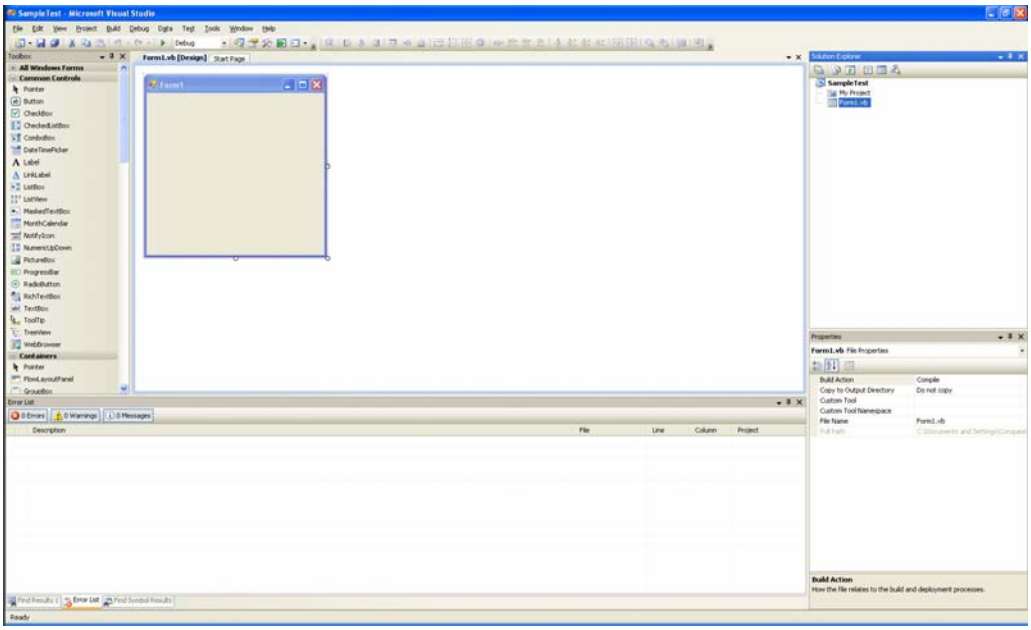
Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

5.1 Setting up your Program

Launch Visual Studio 2008 and create a new VB.NET Forms application by selecting a Visual Basic project type along with the Windows Forms Application project template (NOTE: The SDK can be used with any .NET language such as C#.). Make sure that the .NET Framework is set to 2.0 and give the project the name CV5200_Sample_Project as shown below. Then press the OK button to continue.



Once you click on OK you will have a blank project open that looks similar to the screen below. If the solution explorer is not open (see right hand side of image below), you will need to open it by clicking ***View … Solution Explorer*** on the Visual Studio menu.

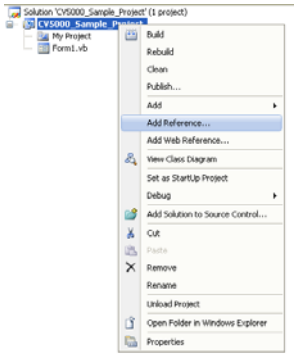


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

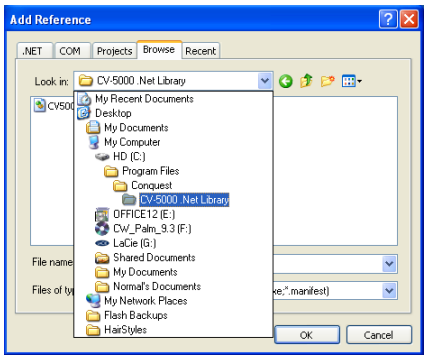
5.2 Adding a Reference to the CV-5000 .NET Library

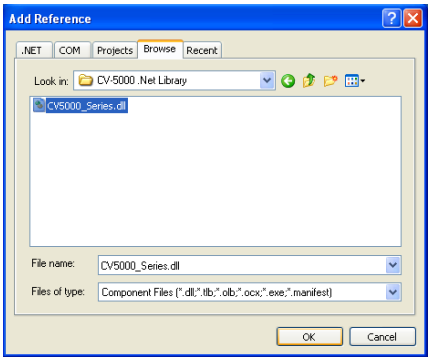
Before you can use the CV-5000 .NET Library in your application you must add a reference to it in your code. This will allow you to access the properties and methods of the CV-5000 .NET Library.

1. In the Solution Explorer, right click on CV5000_Sample_Project and select Add Reference…



2. Click on the browse tab and navigate to the directory where the CV-5000 .NET Library was installed. The default directory is C:\Program Files\Conquest\CV-5000 .Net Library. Select CV5000_Series.dll and then click OK.



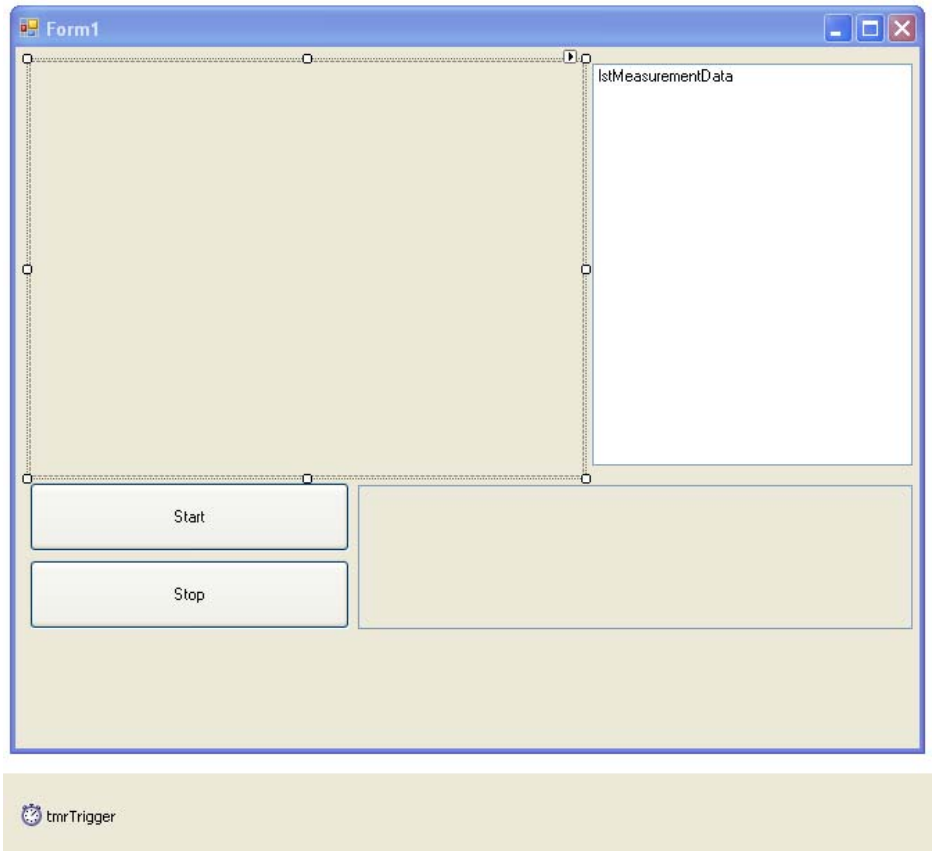


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

5.3 Add Required GUI Components to the Application's Main Form

1. Add a PictureBox to the form and rename it to picPartImage. Set the size mode to StretchImage and place in the top left of the form.
2. Add a ListBox to the form and rename it lstMeasurementData. Place in the top right of the form.
3. Add a Button to the form and rename it to btnStart. Set the Text Property to "Start".
4. Add another button to the form and rename it to btnStop. Set the Text Property to "Stop".
5. Add Timer control to the form and rename it to tmrTrigger.
6. Add a TextBox to the form and rename it to txtStatus. Set the ReadOnly property and the Multiline property to true. Place txtStatus in the bottom right of the form.

After all the controls are laid out, the form should look similar the following image:



Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

5.4 Connecting to the CV-5000 Device

In this section we will be adding code to the application, the light gray code is code that has already been added to the application either by the windows IDE or in a previous step and is just included to help the user grasp where the code is to be added. In this section we will add the code to connect to the CV-5000 device and display the connection status on the screen so the user can see that they are connected. This section will also introduce the basic concepts of thread-safe programming and handling the events that are raised by the CV-5000 Library. The sections that follow will build on these basic concepts to retrieve measurement data and finally an image from the CV-5000.

5.4.1 Add CV5000_Series Namespace & Create an CV-5000 Library Object

Use the *Imports* statement to enable access to the CV-5000 library's namespace. Before you add the *Imports* statement you must have previously added a reference to the library as defined in Section 5.2. Now double click on the title of the form to bring up the forms code window and add the following above the forms class definition as shown:

```
Imports CV5000_Series

Public Class Form1

      Private Sub Form1_Load(ByVal sender As System.Object, ByVal e _
      As System.EventArgs) Handles MyBase.Load

      End Sub
End Class
```

For this sample program we are going to create a CV-5000 object that is global to Form1. Add the following code just below the forms class definition as shown below:

```
Imports CV5000_Series

Public Class Form1

    Private _cv5000 As CV5000
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

5.4.2 Connecting to the CV

The sample program will connect to the CV-5000 as soon as it starts up so we will need to create an instance of the CV-5000 library and configure it to connect to the CV-5000 when the form loads. This is accomplished by adding the following code to the form's load event as shown below:

```
Private Sub Form1_Load(ByVal sender As System.Object, _
      ByVal e As System.EventArgs) Handles MyBase.Load

  'Create an instance of the CV-5000 library
  _cv5000 = New CV5000()


  'Set up the CV-5000 I.P. address and port information
  _cv5000.RemotePort = 8500
  _cv5000.RemoteImagePort = 8501
  _cv5000.RemoteHost = "192.168.168.211"

  'Connect to the CV-5000 series device
  _cv5000.Connect()

End Sub
```

NOTE - For the RemoteHost property, use the IP address that was configured in the CV-5000 series device being connected to.

NOTE - The CQActivation.dll must be copied over to the application directory for the program to run. This directory will be named Debug or Release depending on what configuration is set with the configuration manager in Visual Studio. See "section 4.5 Deployment" for more information on this.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

5.4.3 Creating Event Handlers

In the last section you created a connection to the CV-5000 but we do not know whether or not the connection was successful. This is because the CV-5000 .NET Library communicates with the application through events. A complete list of the events is contained in Section 6.2. The sample application will use a subset of these events. It is through these events that your application will receive the data and any status information that the CV-5000 returns.

To set up your code to be able to handle these events you first have to create "hooks" into the CV-5000 .NET Library's events and then you have to create the event handlers in your application. Since we want to be able to get feedback from the Connect method we must add these hooks before we connect to the CV-5000. Add the following code to the form's load method as shown below.

```
'Set up the CV-5000 I.P. address and port information
_cv5000.RemotePort = 8500
_cv5000.RemoteImagePort = 8501
_cv5000.RemoteHost = "192.168.168.211"

'Link our handlers to the events being created by the CV-5000
AddHandler _cv5000.ConnectionEstablished, _
        AddressOf ConnectionEstablished
AddHandler _cv5000.ImageConnectionEstablished, _
        AddressOf ImageConnectionEstablished
AddHandler _cv5000.ErrorOccurred, _
        AddressOf ErrorOccurred

'Connect to the CV-5000 series device
_cv5000.Connect()
```

Now that the hooks are in place we can add the event handlers. Add the following methods to the form's class definition. The `SetStatus` method is defined in the next section.

```
Public Class Form1

    'Triggered when the command port is connected to
    Private Sub ConnectionEstablished()
        SetStatus("Command port connected")
    End Sub

    'Triggered when the image port is connected to
    Private Sub ImageConnectionEstablished()
        SetStatus("Image port connected")
    End Sub

    Private Sub ErrorOccurred(ByVal args As ErrorOccurredEventArgs)
        If Not args.ExceptionCaught Is Nothing Then
            MessageBox.Show(args.ErrorMessage + "," + _
                            args.ExceptionCaught.ToString())
        Else
            MessageBox.Show(args.ErrorMessage)
        End If
    End Sub

End Class
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

5.4.4 Updating the Status Textbox in a Thread-safe Manner

The CV-5000 .NET Library uses independent threads to poll for data from the CV-5000 device. Because of this, it is necessary to create your application in a thread-safe manner. Since Windows forms controls are not thread-safe the controls on the form cannot be directly updated from the event handlers in the application. Therefore the creation of a thread-safe method is required to update any of the controls on the form.

The sample application has a status field that is updated whenever a significant event has happened. The code below shows how to use the CV-5000 .NET Library in a thread-safe way to update the status field in the application. Add the following methods to the form's class.

```
Public Class Form1

    'Delegate defines function used to update GUI in a threadsafe manner
    Delegate Sub UpdateStringDelegate(ByVal entry As String)

    'Called with an invoke when udating txtStatus in event handler
    Private Sub UpdateStatus(ByVal entry As String)
        txtStatus.Text = entry + vbCrLf + txtStatus.Text
    End Sub

    'Update status TextBox, use invoke if necessary
    Private Sub SetStatus(ByVal entry As String)
        If InvokeRequired Then
            'Not thread safe call, update control in a thread safe manner
            Invoke(New UpdateStringDelegate(AddressOf UpdateStatus), entry)
        Else
            'Thread safe call, update control directly
            UpdateStatus(entry)
        End If
    End Sub

End Class
```

The `SetStatus` method checks a flag "`InvokeRequired`" that is supplied by the .NET Framework which lets the user know if the event was raised in the user-interface thread or not. If the event was raised in the user interface thread (`InvokeRequired` = False) then we can call the `UpdateStatus` method directly otherwise we have to use the delegate method `UpdateStringDelegate` to call the `UpdateStatus` method.

As you can see from the code above the `UpdateStringDelegate` defines a method signature that can be used with the Invoke Method to update a Windows Forms GUI control and its signature is compatible with the `UpdateStatus` method. To boil the Microsoft tech-speak into English what this means is that the `UpdateStringDelegate` method and the `UpdateStatus` method have the same return type (in this case none) and the same parameter list.

NOTE - See the following article in Microsoft's documentation for more information on this concept. http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invokerequired.aspx

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

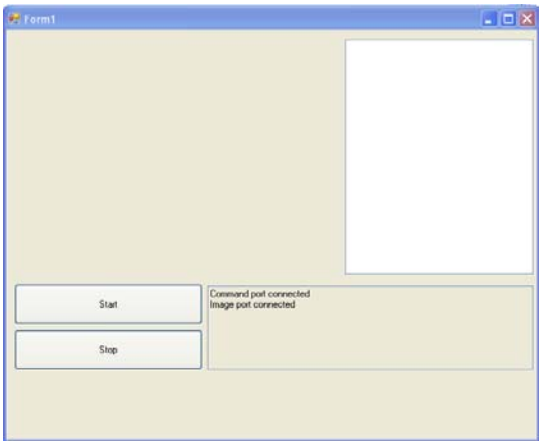
5.4.5 Closing Connections

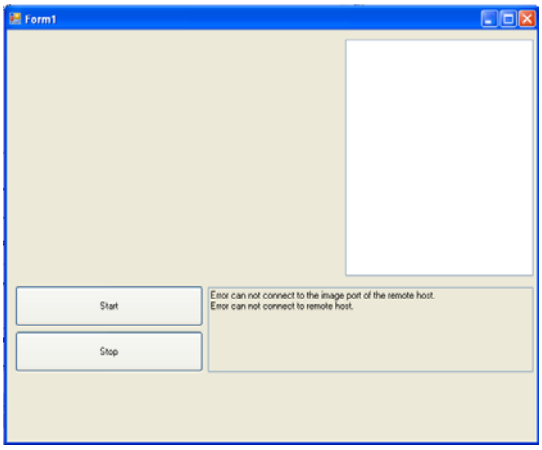
Before the application shuts down, you should close your connection to the CV-5000. To do this the CloseConnection() method in the CV-5000 .NET Library should be called. Add a form closing method to the sample application and add a call to CloseConnection() as shown below.

```
Public Class Form1

    Private Sub Form1_FormClosing(ByVal sender As System.Object, _
      ByVal e As System.Windows.Forms.FormClosingEventArgs) _
      Handles MyBase.FormClosing
            _cv5000.CloseConnection()
    End Sub

End Class
```

5.4.6 Running the Application with Connection Events

In its present state, all the application does is connect to the CV-5000 series device. Run the application, since the connect command is in the Form1_Load event handler, once you run the program you should see something similar to the image on the left below. (It may take a second or two for the "command port connected" text to appear) If the application is not able to connect to the CV-5000 series device, the error message(s) shown in the image on the right will appear after several seconds.





If you ran into an error while creating the code in this section, Appendix A.1 contains a listing of the code up to this point.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

5.5 Retrieving Measurement Data from the CV-5000 Device

In this section we will be expanding the code that we created in section 5.4 to retrieve measurement data from the CV-5000 and display it in the measurement list box. We will introduce a new event (DataReceived) and set up the Start and Stop buttons.

5.5.1 Updating the Measurement ListBox in a ThreadSafe way

The sample application has a measurement listbox that is updated each time measurement data is received. The code below shows how to use the CV-5000 .NET Library in a thread-safe way to update the measurement listbox in the application. Add the following methods to the form class.

```
Public Class Form1

    'Add data received to measurement list box
    Private Sub UpdateMeasurementData(byval entry As String)
        lstMeasurementData.Items.Add(entry)
    End Sub

    'Update Measurement list box, use invoke if necessary
    Private Sub SetMeasurementData(ByVal entry As String)
        If InvokeRequired Then
            'Not thread safe call, update control in a thread safe manner
            Invoke(New UpdateStringDelegate(AddressOf UpdateMeasurementData) _
                    ,entry)
        Else
            UpdateMeasurementData(entry)
        End If
    End Sub

End Class
```

The `SetMeasurementData` method checks a flag "`InvokeRequired`" that is supplied by the .Net Framework which lets the user know if the event was raised in the user-interface thread or not. If the event was raised in the user interface thread (`InvokeRequired` = False) then we can call the `UpdateMeasurementData` method directly otherwise we have to use the delegate method to call the `UpdateMeasurementData` method.

NOTE - The `UpdateStringDelegate` method is the same method that was created in section 5.4.4 so it is not recreated here. You will notice that the `UpdateMeasurementData` method has a compatible signature which is why we can use the same delegate.

NOTE - See the following article in Microsoft's documentation for more information on this concept. http://msdn.microsoft.com/en-us/library/system.windows.forms.control.invokerequired.aspx

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

5.5.2 Add Data Received Event Handler

When the CV-5000 .NET Library receives data from the CV-5000 it raises the Data Received event. To retrieve the data you must "hook" into the event and add a handler for it. First hook into the event in the form's load method with the rest of the handlers as shown below:

```
'Link our handlers to the events being created by the CV-5000
AddHandler _cv5000.ConnectionEstablished, _
        AddressOf ConnectionEstablished
AddHandler _cv5000.ImageConnectionEstablished, _
        AddressOf ImageConnectionEstablished
AddHandler _cv5000.ErrorOccurred, _
        AddressOf ErrorOccurred
AddHandler _cv5000.DataReceived, _
        AddressOf DataReceived

'Connect to the CV-5000 series device
_cv5000.Connect()
```

Now that the hooks are in place we can add the event handler to our application. Add the following method to the form class.

```
Public Class Form1

    Private Sub DataReceived(ByVal args As DataEventArgs)
        Dim Str As String

        ' The data from the CV-5000 is stored in 'args'
        If args.ASCIIData.Length > 30 Then
            Str = args.ASCIIData.Substring(0, 30) + "..."
        Else
            Str = args.ASCIIData
        End If

        If args.RecMode = CVRecMode.CVError Then
            SetMeasurementData(Str)
        ElseIf args.Port = CVPort.CVDataPort Then
            SetMeasurementData(Str)
        End If

    End Sub

End Class
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

5.5.3 Setting up the Timer

The timer will be the source of triggers to the CV-5000 series device. Each time the timer expires it will send a trigger to the CV. Double click on the tmrTrigger control to add a handler for the timer event and add the following code to the event.

```
Private Sub tmrTrigger_Tick(ByVal sender As System.Object, _
      ByVal e As System.EventArgs) Handles tmrTrigger.Tick

        _cv5000.Trigger(1)

End Sub
```

5.5.4 Setting up the Start and Stop Buttons

The start and stop buttons on the form will enable and disable the timer on the form. Create event handlers by double clicking on the buttons and add the following code to the application.

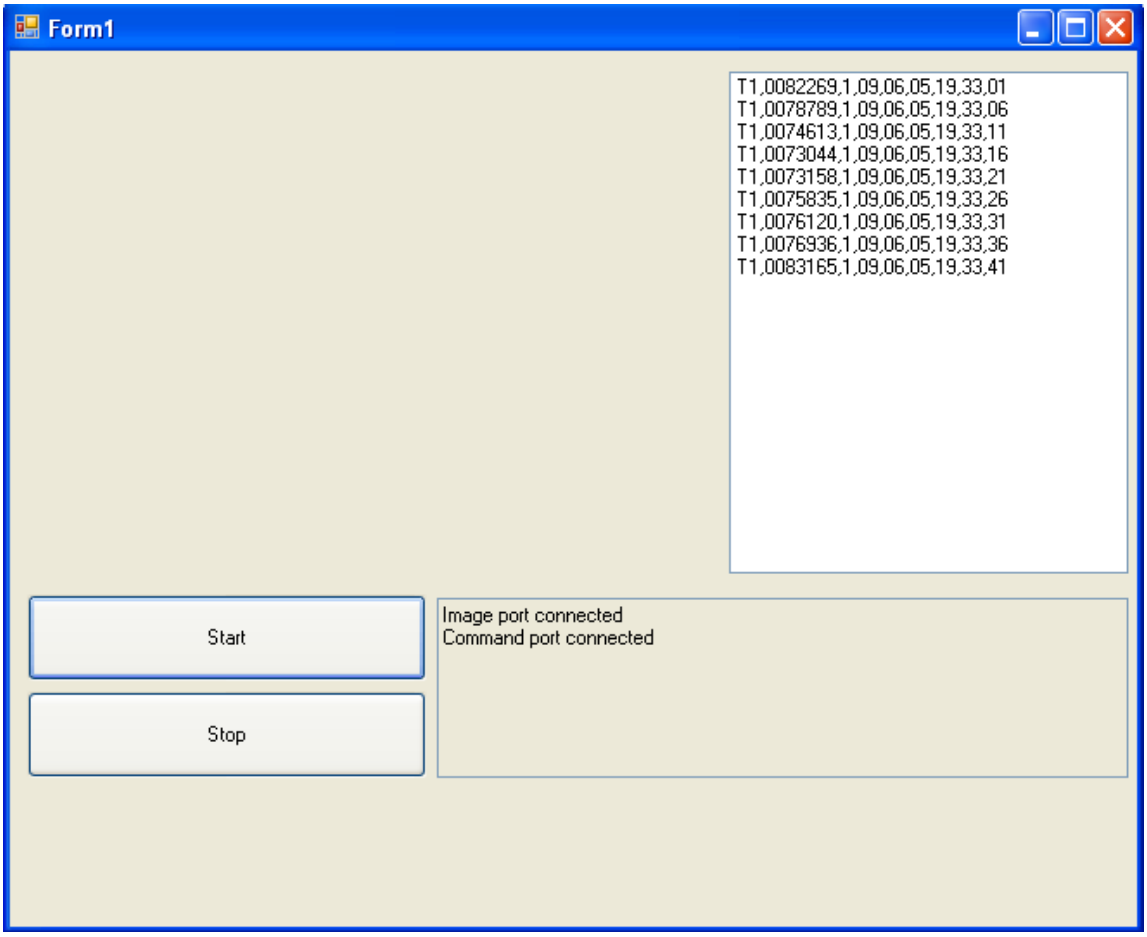
```
Private Sub btnStart_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnStart.Click

    'Trigger the CV-5000 device now
    _cv5000.Trigger(1)

    tmrTrigger.Interval = 5000
    tmrTrigger.Enabled = True

End Sub

Private Sub btnStop_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnStop.Click

    tmrTrigger.Enabled = False

End Sub
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

5.5.5 Running the Application with the Trigger Data

Run the application and press the start button. The output should look similar to the following screen shot on the right. The actual data shown will depend on the configuration of the program in the CV-5000 series device. The data will be received once every 5 seconds so you may want to wait for about 20 seconds or so to elapse.



If you ran into an error while creating the code in this section, Appendix A.2 contains a listing of the code up to this point.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

5.6 Retrieving Image Data from the CV-5000 Device

In this section we will be expanding the code that we created in section 5.5 to retrieve image data from the CV-5000 and display it in the picture box.

5.6.1 Updating the PictureBox in a Thread-safe Manner

The sample application has a picture box that is updated each time image data is received. The code below shows how to use the CV-5000 .NET Library in a thread-safe way to update the picture box in the application. Add the following methods to the form class.

```
Public Class Form1

    Delegate Sub UpdateBitmapDelegate(ByVal bmp As Bitmap)

    'Add image to picturebox
            Private Sub UpdateBitmap(ByVal bmp As Bitmap)Dim oldBitmap As
    Bitmap
        oldBitmap = picPartImage.Image

        If Not oldBitmap Is Nothing Then
           oldBitmap.Dispose()
        End If

        picPartImage.Image = bmp

    End Sub


    'Update picturebox, use invoke if necessary
    Private Sub SetBitmap(ByVal bmp As Bitmap)
        If InvokeRequired Then

            'Not thread safe call, update control in a thread safe manner
            Invoke(New UpdateBitmapDelegate(AddressOf UpdateBitmap),bmp)

        Else

            UpdateBitmap(bmp)

        End If
    End Sub


End Class
```

The `SetBitmap` method checks a flag "`InvokeRequired`" that is supplied by the .Net Framework which lets the user know if the event was raised in the user-interface thread or not. If the event was raised in the user interface thread (`InvokeRequired` = False) then we can call the `UpdateBitmap` method directly otherwise we have to use the delegate method, `UpdateBitmapDelegate`, to call the `UpdateBitmap` method.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

5.6.2 Add Image Data to the Data Received Handler

When the data is received we need to call the correct handler based on the type of data being received, so add the following code to the end of the DataReceived event handler.

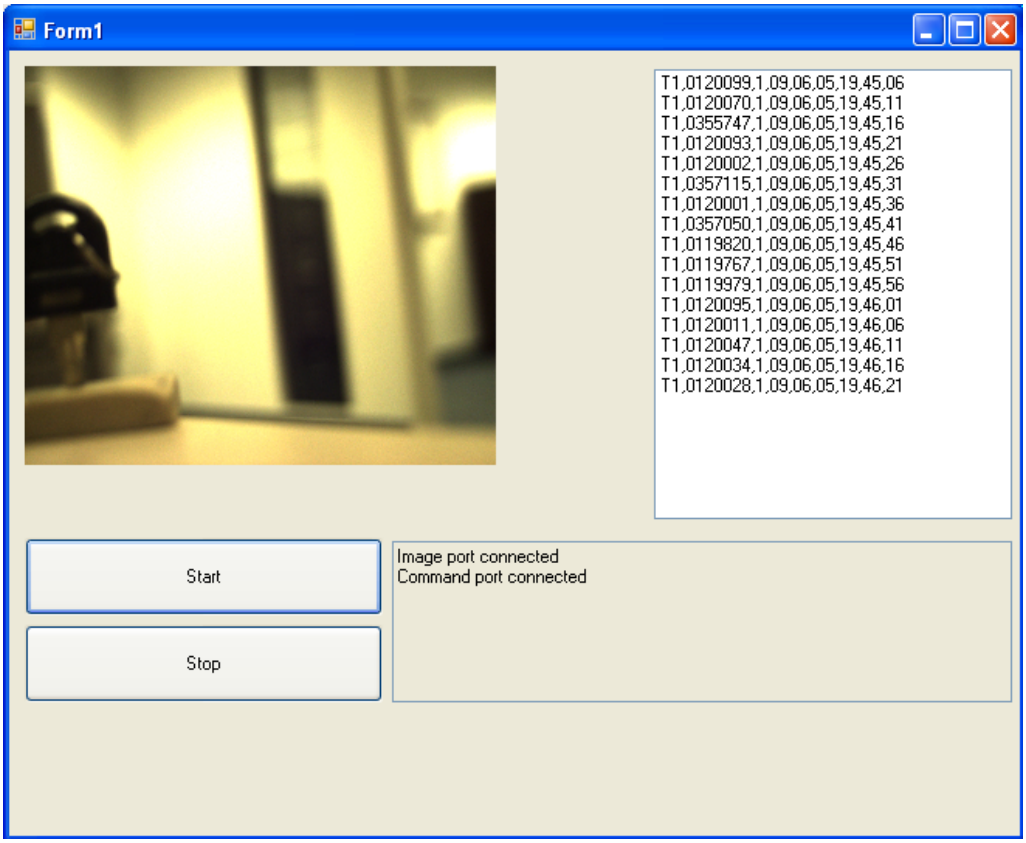
```
Public Class Form1

    Private Sub DataReceived(ByVal args As DataEventArgs)
        Dim Str As String

        If args.ASCIIData.Length > 30 Then
            Str = args.ASCIIData.Substring(0, 30) + "..."
        Else
            Str = args.ASCIIData
        End If

        If args.RecMode = CVRecMode.CVError Then
            SetMeasurementData(Str)
        ElseIf args.Port = CVPort.CVDataPort Then
            SetMeasurementData(Str)
        End If

        If args.RecMode = CVRecMode.TrigImageData _
            Or args.RecMode = CVRecMode.RegImageBMP _
            Or args.RecMode = CVRecMode.ScreenCapture Then
            SetBitmap(args.Bitmap)
        End If

    End Sub

End Class
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

5.6.3 Running the Complete Sample Program

Run the program and the program should display both the measurement data and an image with each trigger. You may want to pass your hand in front of the camera to be sure that the image is changing. Again the image will be triggered once every 5 seconds so you may want to wait a few seconds.



If you ran into an error while creating the code in this section, Appendix A.3 contains a listing of the complete code.

CONGRATULATIONS!!! You have just completed your first program.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6 THE CV-5000 .NET LIBRARY REFERENCE

The CV-5000 .NET Library contains a single .NET Assembly called CV5000_Series.dll. The CV-5000 .NET library provides the complete Ethernet interface to your CV-5000 series device. The library contains properties, methods, and events which the user can use to create a custom CV-5000 application.

6.1 Properties

Most of the CV-5000 .NET Library property values are used in the initial setup of the control.

6.1.1 Property: Name

Type: String

This property can be used for applications that have more than one instance of the CV-5000 class. Once this property is set, the CV-5000's ToString() method will return the instance's name. For example:

```
Private _cv5000 As CV5000
_cv5000.Name = "CVTest"
str = _cv5000.ToString()
```

In the case shown above str would contain "CVTest". An example of where this would be useful is when used with the Windows Forms ComboBox. Simply add all the CV-5000 instances to the ComboBox's Items collection and the CV-5000 instance names will automatically be shown in the ComboBox's list.

6.1.2 Property: Password

Type: String

The CV-5000 .Net Library requires that the developer provide the correct password to unlock the control. Without the password, the control will not function. This password will be provided by Conquest Consulting on the purchase of the CV-5000 .Net Library.

This property is no longer required.

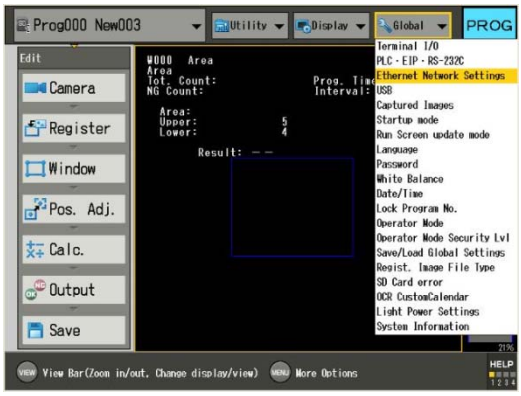
Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

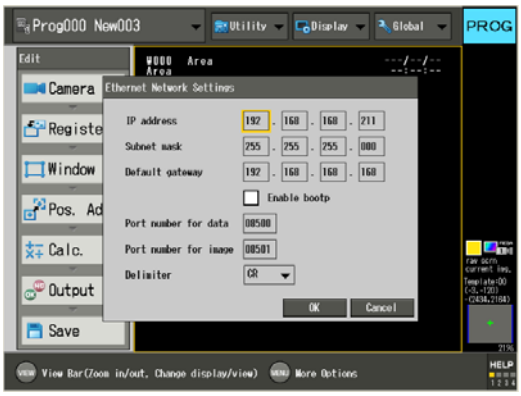
6.1.3 Property: RemoteHost

This property must be set to the IP address of the CV-5000 series device. This property must be set before calling any of the methods.

To change or view the Ethernet settings of your CV-5000 device:

1. While in PROGRAM mode, select Global on the top right of the screen and click on Ethernet Network Settings.
2. You will be able to change or view the Ethernet IP address and/or ports from here.





For more information on connecting your CV-5000 series device, see the CV-5002 Series instruction manual.

6.1.4 Property: RemotePort

Type: Integer

This property must be set to the port number that the CV-5000 series device is currently using for the data transmission. This property must be set before calling any of the methods.

To change or view the Ethernet settings of your CV-5000 see the instructions in the earlier section (Property: RemoteHost).

6.1.5 Property: RemoteImagePort

Type: Integer

This property must be set to the port number that the CV-5000 series device is currently using for the image transmission. This property must be set before calling any of the methods.

To change or view the Ethernet settings of your CV-5000 see the instructions in the earlier section (Property: RemoteHost).

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.1.6 Property: DebugMode_LogToFile

Type: Boolean

This property can be set to True to assist developers in troubleshooting communications errors with the CV-5000 series device. When enabled, the CV-5000 .NET Library will log all communications between the PC and the CV-5000 series device to an ASCII log file given the name 'date.log' where date is the current date. The binary transmissions will be saved in individual files with a .bin extension.

This option tremendously degrades the performance of the CV-5000 and should only be enabled when troubleshooting. This option also uses a lot of disk space because the binary transmissions are logged as well. Be sure this option is set to False before distributing your application.

Debug mode logs can be found in the following directory:

XP-C:\Documents and Settings\Conquest\Local Settings\Application Data\Keyence_CV
Vista-C:\Users\UserName\AppData\Local\Keyence_CV

This option is turned off by default.

6.1.7 Property: ErrorLogging

Type: Boolean

This property can be set to True to assist developers in diagnosing errors with the CV-5000 series device. When enabled, the CV-5000 .NET Library will log all errors that occur to an ASCII log file given the name 'date.err' where date is the current date.

Error mode logs can be found in the following directory:

XP-C:\Documents and Settings\Conquest\Local Settings\Application Data\Keyence_CV
Vista-C:\Users\UserName\AppData\Local\Keyence_CV

This option is turned on by default.

6.1.8 Property: BinaryDump

Type: Boolean

This property can be set to True to assist developers in diagnosing errors with the CV-5000 series device. When enabled, the CV-5000 .NET Library will all data that comes in from the CV Device and save it to a series of files. (This option can use a large amount of disk space.)

BinaryDump logs for each day will have their own directory name based on the day's date. The binary dump directories can be found in the following directory:

XP-C:\Documents and Settings\Conquest\Local Settings\Application Data\Keyence_CV
Vista-C:\Users\UserName\AppData\Local\Keyence_CV

This option is turned off by default.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.2 Events

The CV-5000 .NET Library follows an event driven model. All communication from the CV-5000 .NET Library will take place with an event.

6.2.1 Event: ErrorOccurred

This event is fired when there is a problem communicating with the CV-5000 series device. This event will NOT be fired if the CV-5000 device rejects the command and returns an Error Code. If the CV-Device returns an error code, that error code will be returned with a DataReceived event.

The event arguments for the ErrorOccurred event are defined by the **ErrorOccurredEventArgs** class. The ErrorOccurredEventArgs class has the following public properties:

ErrorMessage: Type: String - description of the error that occurred.

ExceptionCaught: Type: Exception - this property contains the Exception that caused the communications failure to occur. If there was no exception involved in the error then the value of this property will be NULL.

ErrorType: Type: CVErrorType Enumeration – this value can be used to help pinpoint the cause of the error by indicating what the library was doing when the error occurred. The CVErrorType enumeration will contain one of the following values:

Test_Error,

Connect_Error,

ImageConnectCallback_Error,

ConnectCallback_Error,

Disconnect_Error,

SendRawCmd_Error,

SendRawCmdArray_Error,

SendRawCmdCallback_Error,

ResetCommand_Error,

Trigger_Error,

ReadImage_Error,

ReceiveDate_Error,

GetImageData_Bitmap_Error,

RetrieveImages_Error,

ChangeCurrentProgramNumber_Error,

Misc_Error,

ReadCurrentProgramNumber_Error,

SwitchDisplayTemplate_Error,

SwitchResultDisplay_Error,

SwitchPage_Error,

SwitchScreen_Error,

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

SwitchScreenNumber_Error,

RepeatOutputMeasurementResult_Error,

ChangeCurrentWindowNumber_Error,

ReadCurrentWindowNumber_Error,

WriteProgramData_Error,

RequestProgramData_Error,

RequestCurrentProgramData_Error,

WriteEnvironmentalSettings_Error,

RequestEnvironmentalSettings_Error,

RetrievingListOfLibraryNumbers_Error,

WriteBinaryFilterData_Error,

ReadBinaryFilterData_Error,

RegisterImage_Error,

RequestImageData_Error,

RequestImageDataToSDCard_Error,

ChangeModeToRUN_Error,

ChangeModeToProgram_Error,

RequestScreenCapture_Error,

SaveScreenCaptureImage_Error,

RetrieveListRegisteredImages_Error,

ChangeShutterSpeed_Error,

ChangeCameraSensitivity_Error,

SetTriggerDelay_Error,

SetLightIntensityValue_Error,

SetExtensionPatternCounter_Error,

ReadImageBufferWriteToSDCard_Error,

ImageDataWorker_Error,

AnalyzeHeaderBytes_Error,

WriteRegisteredImage_Error,

ReadRegisteredImage_Error,

WriteLibraryData_Error,

ReadLibraryData_Error,

WriteGlobalSettings_Error,

ReadGlobalSettings_Error,

WriteWindowLimit_Error,

ReadWindowLimit_Error,

WriteToCommandMemory_Error,

SetInitialValuesForCommandMemory_Error,

ReadFromCommandMemory_Error,

SetDateTime_Error,

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

ReadDateTime_Error,

Reset_Error,

RecalculateReferenceValue_Error,

SaveProgramSettings_Error,

TriggerInputEnable_Error,

TriggerInputDisable_Error,

EnableCommunicationPort_Error,

DisableCommunicationPort_Error,

ChangePassword_Error,

RemoteControlPseudoInput_Error,

EchoText_Error,

SaveStatisticalData_Error,

CommandWorker_Error,

ImageWorker_Error

6.2.2 Event: ConnectionEstablished

This event is fired when the application successfully connects with the data port of the CV-5000 series device. If communication is attempted before a socket connection is established with the CV-5000 series device then an ErrorOccurred event will be fired. The host application should monitor this event to insure that communication with the CV-5000 series device has been established.

This event has no event arguments.

6.2.3 Event: ImageConnectionEstablished

This event is fired when the application successfully connects with the image port of the CV-5000 series device. The host application should monitor this event to insure that images can be received from the CV-5000 series device.

This event has no event arguments.

6.2.4 Event: DataSent

This event is fired when the application sends a command to the CV-5000 series device. This event will be fired if the application was able to send a command to the CV-5000 series device. Receiving this event does not indicate that the CV-5000 device actually received the transmission. Monitor the DataReceived event for confirmation that the command was received by the CV-5000 series device.

The event arguments for the DataSent event are defined by the **DataSentEventArgs** class. The DataSentEventArgs class has the following public properties:

Data: Type: String – Command that was sent to the CV-5000 Series Device.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.2.5 Event: DataReceived

This event is fired when the application receives data from the CV-5000 Series Device on either the data port or the image port. This event can be triggered as a response to a command sent to the CV-5000 Series device or an unsolicited communication from the CV-5000 series device to the application.

The event arguments for the DataReceived event are defined by the **DataEventArgs** class. The DataEventArgs class has the following public properties:

RawData Property: Type: Array of Bytes – This property will return the entire packet of data received by the CV-5000 series device. Most applications will not need to use this information. The other properties in the DataEventArgs class will provide more useful information to the host application.

CommandCode Property: Type String – This property contains the command code received (two character string) for the data received from the CV-5000 series device. This command code can be used by the application to match up commands sent to the CV-5000 series device with the responses received from the CV-5000 series device.

CommandCode2 Property: Type String – This property is used if the CommandCode property was “ER”. In this case the CommandCode2 property will contain the command sent to the CV-5000 series device that resulted in an error response from the device.

DataType Property: Type: CVDataType - This property indicates the type of data returned from the CV-5000 series device. The CVDataType enumeration will contain one of the following values:

- ASCII
- Binary
- Error
- Unknown

If the DataType property contains a value of ‘ASCII’ or a value of ‘Binary’ then no errors happened and the DataEventArgs object contains valid data. If the CV-Device rejected the command the DataType property will contain a value of ‘Error’. The ErrorOccurred event will NOT be triggered in this case. Under normal operation, the DataType property will not contain a value of ‘Unknown’.

RecMode Property: Type: CVRecMode – This property is used to interpret the data received from the CV-5000 series device. Whenever data is received from the CV-5000 series device, this device must be checked so the host application will know what properties to look at in the DateEventArgs object. For example, if the value of CVRecMode is TrigImageData then the application will know that the Bitmap property contains a valid bitmap. The CVRecMode enumeration will contain one of the following values:

Unknown

Norm

ScreenCapture

ImageData

ProgData

EnvSettings

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

RegImageBMP

RegImageJPEG

TrigResultData

TrigImageData

TrigNoData

CVError

ListRegImages

ListLibraryNumbers

CmdResponseNoData

CmdScreenCapture

ProgramNumber

WindowNumber

SavedRegImageData

FilterData

GlobalSettings

WindowLimit

CommandMemory

DateTime

Echo

Port Property: Type: CVPort – This property indicates whether the data received came over the data port or the image port. CVPort is a public enumeration in the CV-5000 series library and the two possible values are 'CVDataPort' and 'CVImagePort'.

BinaryData Property: Type: byte Array – This property will contain an array of bytes if the DataType property indicates a response of binary data. Otherwise this property will be NULL.

ASCIIData Property: Type: String – This property will return the data received as a string from the first byte received until the first carriage return in the data received.

ErrorCode Property: Type CVErrorCode – If the DataType property indicates an 'Error' then this property should be read by the application to determine what the error is. The CVErrorCode Enumeration can contain one of the following values:

No_Error = -1

Invalid_Command = 0

Command_Action_Disabled = 1

Work_Memory_Error = 2

Parameter_Error = 22

Password_Error = 80

Password_Command_Error = 81

Checksum_Error = 90

Timeout_Error = 91

ErrorCodeString Property: Type string – Returns the string representation of the value in the ErrorCode property with the underscores removed.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

Bitmap Property: Type Bitmap – This property will contain a bitmap if a bitmap was received from the CV-5000 series device. The CVRecModes that indicate a bitmap are 'ScreenCapture', 'RegImageBMP', 'RegImageJPEG', and 'TrigImageData'. The property generates a new bitmap. Once the application no longer needs the bitmap, the Dispose() function should be called on the bitmap to release its resources back to windows.

TriggeredImageCameraNumber Property: Type int. Returns the camera image when a triggered image is received. Only valid when receive mode is TrigImageData. This property is only implemented in version 2.0 and later of the CV-5000 library.

TriggeredImageTotalStatus Property: Type TotalStatus. Returns the total status when a triggered image is received. Only valid when receive mode is TrigImageData. This property is only implemented in version 2.0 and later of the CV-5000 library. The TotalStatus enumeration can contain one of the following values.

Not_Applicable = -1

OK = 0

NG = 1

TriggeredImageMeasurementCount Property: Type int. Returns the measurement count when a triggered image is received. Only valid when receive mode is TrigImageData. This property is only implemented in version 2.0 and later of the CV-5000 library.

ResultData Property: Type string – This property is valid for commands that have a DataType of 'ASCII'. The meaning of this property will depend on the command that was sent to the CV-5000 device. See the documentation for the command sent to the CV-5000.

Result Property: Type integer – This property will return a numerical value for commands that support it. For commands that don't support a single numerical value being returned, this property will return 0.

6.2.6 Event: ConnectionLost

This event is fired if the CV-5000 .NET library detects a lost connection to the CV-5000 series device. Note that the CV-5000 .NET library can only detect a lost connection if it attempts to send data to the CV-5000 series device after the network connection is lost. To test the network connection, send an `EchoText command.`

This event has no event arguments.

6.2.7 Event: ConnectionClosedByProgram

This event is fired after the host application calls the CloseConnection function in the CV-5000 .NET Library.

This event has no event arguments.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.3 Communication Methods

6.3.1 Method: Connect

Method Signature:	public bool Connect();
Parameters:	None
Return Type:	Boolean True: connection attempted False: Error, connection not attempted.
CV Modes:	ALL
Description:	Requests a connection to the CV-5000 series device. This method must be called before any other methods can be called.
Method Error:	ErrorOccurred event triggered.
CV Error:	N/A
On Success:	If a successful connection is made both the ConnectionEstablished and ImageConnectionEstablished events will be triggered.
C# Example:	cv5000.Connect();
VB.NET Example:	_cv5000.Connect()

6.3.2 Method: CloseConnection

Method Signature:	public bool CloseConnection();
Parameters:	None
Return Type:	Boolean True: Connection Closed. False: Error closing connection.
CV Modes:	ALL
Description:	Closes connections to data and image ports. Sends signals to background threads to exit.
Method Error:	ErrorOccurred event triggered.
CV Error:	N/A
On Success:	If connections successfully closed, the OnConnectionClosed event will be triggered.
C# Example:	cv5000.CloseConnection();
VB.NET Example:	_cv5000. CloseConnection()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.3.3 Method: SendRawCommand

Method Signatures:	public void SendRawCommand(string cmd); public void SendRawCommand(string cmd, byte[] bytes)
Parameters:	cmd: Command in string format to send to CV-5000 series device. bytes: Array of bytes to send to the CV-5000 Series device directly after cmd.
Return Type:	None
CV Modes:	ALL
Description:	This method will send a command to the CV-5000 series device. For a pure ASCII command, use the version of the function with just a cmd parameter. For a command that sends binary data use the version with 2 parameters. Send the ASCII part of the command including the final comma in the cmd parameter. Then send the bytes to be sent in the bytes parameter. The SendRawCommand function will calculate the checksum and append it to the end of the byte array.
Method Error:	ErrorOccurred event triggered.
CV Error:	N/A
On Success:	If data is sent over the network connection, the DataSent event will be triggered but this does not guarantee that the data was received by the CV-5000 series device. Look for an occurrence of the DataReceived event in response to the command sent to the CV-5000 series device.
C# Example:	cv5000. SendRawCommand("T1");
VB.NET Example:	_cv5000. SendRawCommand("T1")

6.3.4 Method: SendRawCommandArray

Method Signature:	public bool SendRawCommandArray(Byte[] sendBytes);
Parameters:	sendBytes: Entire command as an array of bytes.
Return Type:	None
CV Modes:	ALL
Description:	This method will send a command to the CV-5000 series device. Most applications will not need to use this command. Use the SendRawCommand methods.
Method Error:	ErrorOccurred event triggered.
CV Error:	N/A
On Success:	If data is sent over the network connection, the DataSent event will be triggered but this does not guarantee that the data was received by the CV-5000 series device. Look for an occurrence of the DataReceived event in response to the command sent to the CV-5000 series device.
C# Example:	cv5000.SendRawCommandArray(cmdArray);
VB.NET Example:	_cv5000. SendRawCommandArray(cmdArray)

Note – cmdArray is an array of the commands that can be sent to the CV-5000

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4 CV Commands

The following table shows the commands available in the CV-5000 Series and the applicable commands in the CV-5000.NET Library:

CV-5000 Series Command	Code	Functions implemented in CV-5000 .NET Lib
Trigger 1	T1	Trigger()
Trigger 2	T2	Trigger()
Changing to RUN	R0	ChangeModeToRUN()
Changing to Program mode	S0	ChangeModeToProgram()
Repeat output measurement results	M0	RepeatOutputMeasurementResult()
Writing the program number	PW	SwitchProgramNumber()
Reading the program number	PR	ReadCurrentProgramNumber()
Writing the window number	UW	SwitchWindowNumber()
Reading the window number	UR	ReadCurrentWindowNumber()
Display switching	DS	SwitchDisplayTemplate() SwitchResultDisplay() SwitchPageNumber() SwitchPageNext() SwitchPagePrevious() SwitchScreen()
Screen capture	BC	SaveScreenCaptureImage() RequestScreenCapture()
Image registration	BS	RegisterImage()
Retrieving a list of registered images	BL	RetrieveListRegisteredImagesCurrentProgram() RetrieveListRegisteredImages()
Reading the image data	BR	RequestImageData() RequestImageDataToSDCard()
Reading all screen data	BT	ReadAllImageDataInBuffer()
Writing the program data	SW	WriteProgramDataToSDDevice() WriteProgramDataToCurrentProgram() WriteRegisteredImageToSDDevice() WriteRegisteredImageToCurrentProgram() WriteLibraryDataToSDDevice() WriteLibraryDataToCurrentProgram()
Reading the program data	SR	ReadProgramDataFromSDDevice() ReadProgramDataFromCurrentProgram() ReadRegisteredImageFromSDDevice() ReadRegisteredImageFromCurrentProgram() ReadLibraryDataFromSDDevice() ReadLibraryDataFromCurrentProgram()
Writing the Global settings data	SB	WriteGlobalSettingsData()
Reading the Global settings data	SA	ReadGlobalSettingsData()
Retrieving a list of library numbers	DL	RetrieveListOfLibraryNumbers()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Writing the binary data	JW	WriteBinaryFilterData()
Reading the binary data	JR	ReadBinaryFilterData()
Writing the limit settings	DW	WriteWindowCalculationValue()
Reading the limit settings	DR	ReadWindowCalculationValue()
Writing the command memory	MW	WriteToCommandMemory()
Reading the command memory	MR	ReadFromCommandMemory()
Setting initial values for command mem	MI	SetInitialValuesForCommandMemory()
Writing the camera / Light extension unit settings	CW	ChangeShutterSpeed() ChangeCameraSensitivity() SetTriggerDelayValue() ChangeExtensionUnitLightIntensity() SetExtensionPatternCounter()
Writing the date and time settings	TW	SetDateTime()
Reading date and time settings	TR	ReadDateTime()
Recalculating a base value	RR	RecalculateReferenceValues()
Saving the setup values	SS	SaveProgramSettings()
Resetting	RS	Reset()
Trigger input enabling	TE	TriggerInputEnable() TriggerInputDisable()
Communication enabling	LK	EnableCommunicationsPort() DisableCommunicationsPort()
Password	PS	ChangePassword()
Remote control console pseudo input	KY	RemoteControlPseudoInput()
Echoing	EC	EchoText()
Reading statistics data	ST	SaveStatisticalData()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.4.1 Method: Trigger

Method Signature:	public void Trigger(int number);
Base CV Commands:	T1 or T2
Parameters:	number: Trigger number (1 or 2)
Return Type:	None
CV Modes:	RUN
Description:	Performs a trigger.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “T1” or “T2”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	At least one DataReceived event will be fired for the data received over the CVDataPort. A second DataReceived event may be fired for data received over the CVImagePort depending on how the CV device is configured. *If no measurement data is received then the following properties will be set in the instance of DataEventArgs received over the CVDataPort:* CommandCode - "T1" or "T2" depending on which trigger was triggered. RecMode Property - RecTrigNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration) *If measurement data is received then the following properties will be set in the instance of DataEventArgs received over the CVDataPort:* ResultData - String containing result data from the trigger - See CV-5002 Series manual to interpret the data. CommandCode - "T1" or "T2" depending on which trigger was triggered. RecMode Property - TrigResultData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property – ASCII Port Property - CVDataPort (CVPort enumeration) *If image data is received as a result of the trigger, the image data will be received on the CVImagePort. The DataEventArgs will contain the following values:* Bitmap Property - Bitmap of Image BinaryData - Byte array containing bitmap bytes. CommandCode - "T1" or "T2" depending on which trigger was triggered. RecMode Property - TrigImageData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error"

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

DataType Property – Binary

Port Property - CVImagePort (CVPort enumeration)

C# Example: cv5000.Trigger(1);

VB.NET Example: _cv5000.Trigger(1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.4.2 Method: ChangeModeToRun

Method Signature:	public void ChangeModeToRUN();
Base CV Command:	R0
Parameters:	None
Return Type:	None
CV Modes:	PROGRAM
Description:	Changes mode from PROGRAM to RUN.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “R0”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "R0" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. ChangeModeToRUN();
VB.NET Example:	_cv5000. ChangeModeToRUN()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.3 Method: ChangeModeToProgram

Method Signature:	public void ChangeModeToProgram();
Base CV Command:	S0
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Changes mode from RUN to PROGRAM.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “S0”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "S0" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. ChangeModeToProgram();
VB.NET Example:	_cv5000. ChangeModeToProgram()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.4 Method: RepeatOutputMeasurementResult

Method Signature:	public void RepeatOutputMeasurementResult();
Base CV Command:	M0
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Outputs the most recent measurement result again. When automatic image output is set, the image data is also output.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “M0”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	At least one DataReceived event will be fired for the data received over the CVDataPort. A second DataReceived event may be fired for data received over the CVImagePort depending on how the CV device is configured. *If no measurement data is received then the following properties will be set in the instance of DataEventArgs received over the CVDataPort:* CommandCode - "M0" depending on which trigger was triggered. RecMode Property - RecTrigNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration) *If measurement data is received then the following properties will be set in the instance of DataEventArgs received over the CVDataPort:* ResultData - String containing result data from the trigger - See CV-5002 Series manual to interpret the data. CommandCode - "M0" depending on which trigger was triggered. RecMode Property - TrigResultData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property – ASCII Port Property - CVDataPort (CVPort enumeration) *If image data is received as a result of the trigger, the image data will be received on the CVImagePort. The DataEventArgs will contain the following values:* Bitmap Property - Bitmap of Image BinaryData - Byte array containing bitmap bytes. CommandCode - "M0" depending on which trigger was triggered. RecMode Property - TrigImageData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

	ErrorCodeString Property - "No Error"
	DataType Property – Binary
	Port Property - CVImagePort (CVPort enumeration)
C# Example:	cv5000. RepeatOutputMeasurementResult();
VB.NET Example:	_cv5000. RepeatOutputMeasurementResult()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.4.5 Method: SwitchProgramNumber

Method Signature:	public void SwitchProgramNumber(int programNumber, SDOptions sdOption);
Base CV Command:	PW
Parameters:	programNumber: Program number in CV-5000 series device (0 to 999) sdOption: (SDOptions enumeration) Specifies which SD card program is written to.
Return Type:	None
CV Modes:	RUN
Description:	Closes all the displayed program screens and switches to the specified program number.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “PW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "PW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.SwitchProgramNumber(0, SDOptions.SDCard1);
VB.NET Example:	_cv5000.SwitchProgramNumber(0, SDOptions.SDCard1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.6 Method: ReadCurrentProgramNumber

Method Signature:	public void ReadCurrentProgramNumber();
Base CV Command:	PR
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Requests the current program number from the CV-5000 series device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “PR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "PR" SDCard - SDCard1 or SDCard2 RecMode Property - ProgramNumber (CVReceiveMode enumeration) ResultData Property – Three digit string representing the program number. Result Property – Integer representing the program number. ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ReadCurrentProgramNumber();
VB.NET Example:	_cv5000.ReadCurrentProgramNumber()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.7 Method: SwitchWindowNumber

Method Signature:	public void SwitchWindowNumber(int windowNumber);
Base CV Command:	UW
Parameters:	programNumber: Window number in CV-5000 series device (0 to 127)
Return Type:	None
CV Modes:	RUN
Description:	Switches to the specified windows number and refreshes the screen.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “UW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "UW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.SwitchWindowNumber(0);
VB.NET Example:	_cv5000.SwitchWindowNumber(0)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.8 Method: ReadCurrentWindowNumber

Method Signature:	public void ReadCurrentWindowNumber();
Base CV Command:	UR
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Requests the current window number from the CV-5000 series device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “UR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "UR" Result Property - Returned window number (0 - 127). Returns 0 if no window is set. RecMode Property - WindowNumber (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ReadCurrentWindowNumber();
VB.NET Example:	_cv5000.ReadCurrentWindowNumber()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.9 Method: SwitchDisplayTemplate

Method Signature:	public void SwitchDisplayTemplate(DisplaySwitchingTemplateNumber template);
Base CV Command:	DS,PT
Parameters:	template: Value in DisplaySwitchingTemplateNumber Enumeration. The value can be one of the following:
Raw_Screen	
	Filtered_Screen_1 Filtered_Screen_2 Contrast_View User_Defined_Template_1 User_Defined_Template_2 User_Defined_Template_3 User_Defined_Template_4 User_Defined_Template_5 User_Defined_Template_6
Return Type:	None
CV Modes:	RUN
Description:	Switches the display template.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “DS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort:
	CommandCode - "DS"
	RecMode Property - CmdResponseNoData (CVReceiveMode enumeration)
	ErrorCode Property - No_Error (CVErrorCode enumeration)
	ErrorCodeString Property - "No Error"
	Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.SwitchDisplayTemplate(DisplaySwitchingTemplateNumber.SDCard1);
VB.NET Example:	_cv5000.SwitchDisplayTemplate(_ DisplaySwitchingTemplateNumber.SDCard1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.10 Method: SwitchResultDisplay

Method Signature:	public void SwitchResultDisplay(ResultSwitchingDisplayNumber template);
Base CV Command:	DS,RS
Parameters:	template: Value in ResultSwitchingDisplayNumber Enumeration. The value can be one of the following:
None	Default_1 Default_2 Calculation_Results Custom_Display
Return Type:	None
CV Modes:	RUN
Description:	Switches the result display.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “DS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "DS" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. SwitchResultDisplay(ResultSwitchingDisplayNumber.None);
VB.NET Example:	_cv5000. SwitchResultDisplay(_ ResultSwitchingDisplayNumber.None)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.11 Method: SwitchPageNumber

Method Signature:	public void SwitchPageNumber(int pageNumber);
Base CV Command:	DS,PG
Parameters:	pageNumber: 0 to 9999
Return Type:	None
CV Modes:	RUN
Description:	Switches to the specified page.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “DS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "DS" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. SwitchPageNumber(0);
VB.NET Example:	_cv5000. SwitchPageNumber(0)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.12 Method: SwitchPageNext

Method Signature:	public void SwitchPageNext();
Base CV Command:	DS,PG
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Displays the next page.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “DS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "DS" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. SwitchPageNext();
VB.NET Example:	_cv5000. SwitchPageNext()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.13 Method: SwitchPagePrevious

Method Signature:	public void SwitchPagePrevious();
Base CV Command:	DS,PG
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Displays the previous page.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “DS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "DS" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. SwitchPagePrevious();
VB.NET Example:	_cv5000. SwitchPagePrevious()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.14 Method: SwitchScreen

Method Signature:	public void SwitchScreen(int screenNumber);
Base CV Command:	DS,FC
Parameters:	screenNumber: 1 to 5
Return Type:	None
CV Modes:	RUN
Description:	Switches to the specified screen when multiple screens are displayed.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “DS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "DS" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. SwitchScreen(1);
VB.NET Example:	_cv5000. SwitchScreen(1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.15 Method: SaveScreenCaptureImage

Method Signature:	public void SaveScreenCaptureImage();
Base CV Command:	BC,CF
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Saves the current image to the SD Card 2. (Run or Program mode)
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “BC”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "BC" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. SaveScreenCaptureImage();
VB.NET Example:	_cv5000. SaveScreenCaptureImage()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.16 Method: RequestScreenCapture

Method Signature:	public void RequestScreenCapture();
Base CV Command:	BC,CM
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Requests a screen shot and retrieves the image on the CVCommand Port.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “BC”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "BC" Bitmap Property - Windows Bitmap of Image RecMode Property - ScreenCapture (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - Binary Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. RequestScreenCapture();
VB.NET Example:	_cv5000. RequestScreenCapture()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.17 Method: RegisterImage

Method Signature:	public void RegisterImage(int cameraNumber, int registeredImageNumber);
Base CV Command:	BS
Parameters:	cameraNumber: 1 to 4
	registeredImageNumber: 0 to 999
Return Type:	None
CV Modes:	RUN
Description:	Registers the latest image specified by the camera number.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “BS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort:
	CommandCode - "BS"
	RecMode Property - CmdResponseNoData (CVReceiveMode enumeration)
	ErrorCode Property - No_Error (CVErrorCode enumeration)
	ErrorCodeString Property - "No Error"
	Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. RegisterImage(1,0);
VB.NET Example:	_cv5000. RegisterImage(1,0)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.18 Method: RetrieveListRegisteredImagesCurrentProgram

Method Signature:	public void RetrieveListRegisteredImagesCurrentProgram(int cameraNumber);
Base CV Command:	BL,CP
Parameters:	cameraNumber: 1 to 4
Return Type:	None
CV Modes:	RUN
Description:	Retrieves a list of registered images that have been previously saved in the current program.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “BL”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "BL" ResultData - List of Registred images seperated by commas (Example: 000,001,004) RecMode Property - ListRegImages (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. RetrieveListRegisteredImagesCurrentProgram(1);
VB.NET Example:	_cv5000. RetrieveListRegisteredImagesCurrentProgram(1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.19 Method: RetrieveListRegisteredImages

Method Signature:	public void RetrieveListRegisteredImages(int cameraNumber, int programNumber, SDOptions sdOption);
Base CV Command:	BL
Parameters:	cameraNumber: 1 to 4 programNumber: 0 to 999 sdOption: (SDOptions enumeration) SDCard1 or SDCard2
Return Type:	None
CV Modes:	RUN
Description:	Retrieves a list of registered images that have been previously saved to a SD Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “BL”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "BL" ResultData - List of Registred images seperated by commas (Example: 000,001,004) RecMode Property - ListRegImages (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. RetrieveListRegisteredImages(1,0, SDOptions.SDCard1);
VB.NET Example:	_cv5000. RetrieveListRegisteredImages(1,0, SDOptions.SDCard1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.20 Method: RequestImageData

Method Signature:	public void RequestImageData(int cameraNumber, CompressionRatio compressionRatio, SpecifiedImage specifiedImage);
Base CV Command:	BR,CM
Parameters:	cameraNumber: 1 to 4 compressionRatio: (CompressionRatio enumeration) – One,OneHalf, OneQuarter, or OneEigth specifiedImage: (SpecifiedImage enumeration) NW, NG or OK
Return Type:	None
CV Modes:	RUN
Description:	Retrieves a binary image over the CVCommand Port.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “BR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "BR" Bitmap Property - Windows Bitmap of Image BinaryData - Byte array of the Bitmap Image RecMode Property - RegImageBMP (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - Binary Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.RequestImageData(1, CompressionRatio.One, SpecifiedImage.OK);
VB.NET Example:	_cv5000.RequestImageData(1, CompressionRatio.One, SpecifiedImage.OK)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.21 Method: RequestImageToSDCard

Method Signature:	public void RequestImageDataToSDCard(int cameraNumber, CompressionRatio compressionRatio, SpecifiedImage specifiedImage, string fileName);
Base CV Command:	BR,CF
Parameters:	cameraNumber: 1 to 4 compressionRatio: (CompressionRatio enumeration) – One,OneHalf, OneQuarter, or OneEigth specifiedImage: (SpecifiedImage enumeration) NW, NG or OK filename: (Example: /CV/TEST.bmp)
Return Type:	None
CV Modes:	RUN
Description:	The current image data that is stored in the image buffer is written to the SD Card in bitmap format.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “BR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "BR" ResultData = c,t,nnnnnnnnn c: camera number (1 to 4) t: Total Status 0: OK 1: NG nnnnnnnnn: Measurement CountBinaryData - Byte array of the Bitmap Image RecMode Property - SavedRegImageData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.RequestImageDataToSDCard(1, CompressionRatio.One, SpecifiedImage.OK,”/CV/TEST.bmp”);
VB.NET Example:	_cv5000.RequestImageDataToSDCard(1, CompressionRatio.One, SpecifiedImage.OK,”/CV/TEST.bmp”)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.22 Method: ReadAllImageDataInBuffer

Method Signature:	public void ReadAllImageDataInBuffer (int cameraNumber, CompressionRatio compressionRatio, SpecifiedImage specifiedImage, string folderName);
Base CV Command:	BT
Parameters:	cameraNumber: 1 to 4 compressionRatio: (CompressionRatio enumeration) – One,OneHalf, OneQuarter, or OneEigth specifiedImage: (SpecifiedImage enumeration) NW, NG or OK folderName: Name of folder in SD card to save images.
Return Type:	None
CV Modes:	RUN
Description:	All the images that are stored in the image buffer are written to the SD card in bitmap format.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “BT”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "BT" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ReadAllImageDataInBuffer(1, CompressionRatio.One, SpecifiedImage.OK,”/CV”);
VB.NET Example:	_cv5000.ReadAllImageDataInBuffer(1, CompressionRatio.One, SpecifiedImage.OK,”/CV”)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.23 Method: WriteProgramDataToSDDevice

Method Signature:	public void WriteProgramDataToSDDevice(int programNumber, SDOptions option, byte[] programData);
Base CV Command:	SW,DT
Parameters:	programNumber: 0 to 999 sdOption: (SDOption enumeration) SDCard1 or SDCard2 programData: Array of bytes
Return Type:	None
CV Modes:	RUN
Description:	Writes program data to SD Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. WriteProgramDataToSDDevice(programData);
VB.NET Example:	_cv5000. WriteProgramDataToSDDevice(programData)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.24 Method: WriteProgramDataToCurrentProgram

Method Signature:	public void WriteProgramDataToCurrentProgram(byte[] programData);
Base CV Command:	SW,DT,CP
Parameters:	programData: Array of bytes
Return Type:	None
CV Modes:	RUN
Description:	Writes program data to the current program.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. WriteProgramDataToCurrentProgram(programData);
VB.NET Example:	_cv5000. WriteProgramDataToCurrentProgram(programData)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.25 Method: WriteRegisteredImageToSDDevice

Method Signature:	public void WriteRegisteredImageToSDDevice(int programNumber, int cameraNumber, int registeredImageNumber, SDOptions option, byte[] data);
Base CV Command:	SW,BD
Parameters:	programNumber: 0 to 999 cameraNumber: 1 to 4 registeredImageNumber: 0 to 999 option: (SDOption enumeration) SDCard1 or SDCard2 data: Byte array containing registered image data.
Return Type:	None
CV Modes:	RUN
Description:	Writes registered image to SD Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. WriteRegisteredImageToSDDevice(0,1,0, SDOptions.SDCard1,data);
VB.NET Example:	_cv5000. WriteRegisteredImageToSDDevice(0,1,0, _ SDOptions.SDCard1,data)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.26 Method: WriteRegisteredImageToCurrentProgram

Method Signature:	public void WriteRegisteredImageToCurrentProgram(int cameraNumber, int registeredImageNumber, byte[] data);
Base CV Command:	SW,BD,CP
Parameters:	cameraNumber: 1 to 4 registeredImageNumber: 0 to 999 data: Byte array containing registered image data.
Return Type:	None
CV Modes:	RUN
Description:	Writes registered image to the current program.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. WriteRegisteredImageToCurrentProgram(1,0,data);
VB.NET Example:	_cv5000. WriteRegisteredImageToCurrentProgram(1,0,data)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.27 Method: WriteLibraryDataToSDDevice

Method Signature:	public void WriteLibraryDataToSDDevice (int programNumber, int libraryNumber, SDOptions option, byte[] data);
Base CV Command:	SW,DC
Parameters:	programNumber: 0 to 999 libraryNumber: 0 to 999 option: (SDOption enumeration) SDCard1 or SDCard2 data: Byte array containing registered image data.
Return Type:	None
CV Modes:	RUN
Description:	Writes library data to a SD Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. WriteLibraryDataToSDDevice(0,0, SDOptions.SDCard1,data);
VB.NET Example:	_cv5000. WriteLibraryDataToSDDevice(0,0, _ SDOptions.SDCard1,data)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.28 Method: WriteLibraryDataToCurrentProgram

Method Signature:	public void WriteLibraryDataToCurrentProgram(int libraryNumber, byte[] data);
Base CV Command:	SW,DC,CP
Parameters:	libraryNumber: 0 to 999 data: Byte array containing registered image data.
Return Type:	None
CV Modes:	RUN
Description:	Writes library data to the current program.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. WriteLibraryDataToCurrentProgram(0,data);
VB.NET Example:	_cv5000. WriteLibraryDataToCurrentProgram(0,data)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.29 Method: ReadProgramDataFromSDDevice

Method Signature:	public void ReadProgramDataFromSDDevice(int programNumber, SDOptions sdOption);
Base CV Command:	SR,DT
Parameters:	programNumber: 0 to 999
	sdOption: (SDOption enumeration) SDCard1 or SDCard2
Return Type:	None
CV Modes:	RUN
Description:	Retrieves program data from a SD Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort:
	CommandCode - "SR"
	BinaryData Property - Byte array containing program data
	RecMode Property - ProgData (CVReceiveMode enumeration)
	ErrorCode Property - No_Error (CVErrorCode enumeration)
	ErrorCodeString Property - "No Error"
	DataType Property - Binary
	Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. ReadProgramDataFromSDDevice(0,SDOptions.SDCard1);
VB.NET Example:	_cv5000. ReadProgramDataFromSDDevice(0, _ SDOptions.SDCard1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.4.30 Method: ReadProgramDataFromCurrentProgram

Method Signature:	public void ReadProgramDataFromCurrentProgram();
Base CV Command:	SR,DT,CP
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Retrieves program data from the current program.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SR" BinaryData Property - Byte array containing program data RecMode Property - ProgData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - Binary Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. ReadProgramDataFromCurrentProgram ();
VB.NET Example:	_cv5000. ReadProgramDataFromCurrentProgram ()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.31 Method: ReadRegisteredImageFromSDDevice

Method Signature:	public void ReadRegisteredImageFromSDDevice(int programNumber, int cameraNumber, int registeredImageNumber, SDOptions option);
Base CV Command:	SR,BD
Parameters:	programNumber: 0 to 999 cameraNumber: 1 to 4 registeredImageNumber: 0 to 999 option: (SDOption enumeration) SDCard1 or SDCard2
Return Type:	None
CV Modes:	RUN
Description:	Reads registered image from SD Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SR" BinaryData Property - Byte array containing registered image RecMode Property - RegImageJPEG or RegImageBMP (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - Binary Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. ReadRegisteredImageFromSDDevice(0,1,0, SDOptions.SDCard1);
VB.NET Example:	_cv5000. ReadRegisteredImageFromSDDevice(0,1,0, _ SDOptions.SDCard1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.32 Method: ReadRegisteredImageFromCurrentProgram

Method Signature:	public void ReadRegisteredImageFromCurrentProgram(int cameraNumber, int registeredImageNumber);
Base CV Command:	SR,BD,CP
Parameters:	cameraNumber: 1 to 4 registeredImageNumber: 0 to 999
Return Type:	None
CV Modes:	RUN
Description:	Reads registered image from the current program.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SR" BinaryData Property - Byte array containing registered image RecMode Property - RegImageJPEG or RegImageBMP (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - Binary Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ReadRegisteredImageFromCurrentProgram(1,0);
VB.NET Example:	_cv5000. ReadRegisteredImageFromCurrentProgram (1,0)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.33 Method: ReadLibraryDataFromSDDevice (TBD – Update DataEventArgs)

Method Signature:	public void ReadLibraryDataFromSDDevice(int programNumber, int libraryNumber, SDOptions option);
Base CV Command:	SR,DC
Parameters:	programNumber: 0 to 999 libraryNumber: 0 to 999 option: (SDOption enumeration) SDCard1 or SDCard2
Return Type:	None
CV Modes:	RUN
Description:	Reads library data from a SD Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SR" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. ReadLibraryDataFromSDDevice(0,0, SDOptions.SDCard1);
VB.NET Example:	_cv5000. ReadLibraryDataFromSDDevice(0,0, _ SDOptions.SDCard1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.34 Method: ReadLibraryDataFromCurrentProgram (TBD – Update DataEventArgs)

Method Signature:	public void ReadLibraryDataFromCurrentProgram(int programNumber);
Base CV Command:	SR,BD,CP
Parameters:	programNumber: 0 to 999
Return Type:	None
CV Modes:	RUN
Description:	Reads library data from the Current Program.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SR" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. ReadLibraryDataFromCurrentProgram(0);
VB.NET Example:	_cv5000. ReadLibraryDataFromCurrentProgram(0)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.35 Method: RetrieveListOfLibraryNumbers

Method Signature:	public void RetrieveListOfLibraryNumbers(int programNumber, SDOptions sdOption);
Base CV Command:	DL
Parameters:	programNumber: 0 to 999 option: (SDOption enumeration) SDCard1 or SDCard2
Return Type:	None
CV Modes:	RUN
Description:	Retrieves a list of library numbers.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “DL”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "DL" ResultData - List of library numbers seperated by commas (Example: 000,001,004) RecMode Property - ListLibraryNumbers (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. RetrieveListOfLibraryNumbers(0, SDOptions.SDCard1);
VB.NET Example:	_cv5000.RetrieveListOfLibraryNumbers(0, _ SDOptions.SDCard1)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.36 Method: WriteGlobalSettingsData

Method Signature:	public void WriteGlobalSettingsData(byte[] globalSettings);
Base CV Command:	SB
Parameters:	data: Byte array containing global settings.
Return Type:	None
CV Modes:	RUN
Description:	Writes global settings.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SB”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "DB" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.WriteGlobalSettingsData(data);
VB.NET Example:	_cv5000.WriteGlobalSettingsData(data)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.37 Method: ReadGlobalSettingsData

Method Signature:	public void ReadGlobalSettingsData();
Base CV Command:	SA
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Retrieves global settings..
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SA”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SA" BinaryData Property - Byte array containing global settings RecMode Property – GlobalSettings(CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - Binary Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ReadGlobalSettingsData();
VB.NET Example:	_cv5000. ReadGlobalSettingsData()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.38 Method: WriteBinaryFilterData

Method Signature:	public void WriteBinaryFilterData(int windowNumber, int upperLimit, int lowerLimit);
Base CV Command:	JW
Parameters:	windowNumber: 0 to 127 upperLimit: Upper limit of window lowerLimit: Lower limit of window
Return Type:	None
CV Modes:	RUN
Description:	Writes the upper and lower limit on the binary filter of the specified filter.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “JW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "JW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.WriteBinaryFilterData(0,100,10);
VB.NET Example:	_cv5000.WriteBinaryFilterData(0,100,10)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.39 Method: ReadBinaryFilterData

Method Signature:	public void ReadBinaryFilterData(int windowNumber);
Base CV Command:	JR
Parameters:	windowNumber: 0 to 127
Return Type:	None
CV Modes:	RUN
Description:	Obtains binary filter data measurement.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “JR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "JR" ResultData - xxx,yyy (xxx: Upper Limit, yyy: lowerLimit) RecMode Property – FilterData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - Binary Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000. ReadBinaryFilterData(0);
VB.NET Example:	_cv5000.ReadBinaryFilterData(0)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.40 Method: WriteWindowCalculationValue

Method Signature:	public void WriteWindowCalculationValue(int windowNumber, WindowType windowType, string symbol, MeasuredValueLimit measuredValueLimit, double value);
Base CV Command:	DW
Parameters:	windowNumber: 0 to 127 windowType: (WindowType enumeration) MeasurementWindow or CalculationWindow symbol: See page 4-258 in the CV-5002 Series User's Manual measuredValueLimit: (MeasuredValueLimit enumeration) HL or LL value: Value to set calculation value to
Return Type:	None
CV Modes:	RUN
Description:	Writes the window calculation value.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of "ER" and a CommandCode2 of "DW". The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "DW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.WriteWindowCalculationValue(0, WindowType.MeasurementWindow,"X", MeasuredValueLimit.HL,3.0);
VB.NET Example:	_cv5000.WriteWindowCalculationValue(0, _ WindowType.MeasurementWindow,"X", _ MeasuredValueLimit.HL,3.0)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.41 Method: ReadWindowCalculationValue

Method Signature:	public void ReadWindowCalculationValue(int windowNumber, WindowType windowType, string symbol, MeasuredValueLimit measuredValueLimit);
Base CV Command:	DR
Parameters:	windowNumber: 0 to 127 windowType: (WindowType enumeration) MeasurementWindow or CalculationWindow symbol: See page 4-258 in the CV-5002 Series User's Manual measuredValueLimit: (MeasuredValueLimit enumeration) HL or LL
Return Type:	None
CV Modes:	RUN
Description:	Reads the calculation window or limits.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of "ER" and a CommandCode2 of "DR". The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "DR" ResultData - Limit Value as a string RecMode Property - WindowLimit (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ReadWindowCalculationValue(0, WindowType.MeasurementWindow,"X", MeasuredValueLimit.HL);
VB.NET Example:	_cv5000.ReadWindowCalculationValue(0, _ WindowType.MeasurementWindow,"X", _ MeasuredValueLimit.HL)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.42 Method: WriteToCommandMemory

Method Signature:	public void WriteToCommandMemory(int [] commandMemoryPositions, int [] commandMemoryValues);
Base CV Command:	MW
Parameters:	commandMemoryPositions: array of integers commandMemoryValues: array of integers
Return Type:	None
CV Modes:	RUN
Description:	Write to command memory.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “MW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "MW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.WriteToCommandMemory(pos,val);
VB.NET Example:	_cv5000.WriteToCommandMemory(pos,val)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.43 Method: ReadFromCommandMemory

Method Signature:	public void ReadFromCommandMemory(int memoryPosition);
Base CV Command:	MR
Parameters:	memoryPosition: 0 to 999
Return Type:	None
CV Modes:	RUN
Description:	Read value from command memory.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “MR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "MR" ResultData – Memory value as a string RecMode Property - CommandMemory (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ReadFromCommandMemory(0);
VB.NET Example:	_cv5000.ReadFromCommandMemory(0)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.44 Method: SetInitialValuesForCommandMemory

Method Signature:	public void SetInitialValuesForCommandMemory();
Base CV Command:	MI
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Sets all of the current command memory values as the initial values for the command memory.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “MI”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "MI" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.SetInitialValuesForCommandMemory();
VB.NET Example:	_cv5000.SetInitialValuesForCommandMemory()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.45 Method: ChangeShutterSpeed

Method Signature:	public void ChangeShutterSpeed(int cameraNumber, ShutterSpeed shutterSpeed);
Base CV Command:	CW,SH
Parameters:	cameraNumber: 1 to 4 shutterSpeed: (ShutterSpeed enumeration) The value can be one of the following: One_15th One_30th One_60th One_120th One_240th One_500th One_1000th One_2000th One_5000th One_10000th One_20000th
Return Type:	None
CV Modes:	RUN
Description:	Sets all of the current command memory values as the initial values for the command memory.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “CW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "CW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ChangeShutterSpeed(1,ShutterSpeed.one_120th);
VB.NET Example:	_cv5000.ChangeShutterSpeed(1, ShutterSpeed.one_120th)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.4.46 Method: ChangeCameraSensitivity

Method Signature:	public void ChangeCameraSensitivity(int cameraNumber, int sensitivity);
Base CV Command:	CW,SE
Parameters:	cameraNumber: 1 to 4
	sensitivity: 10 to 90
Return Type:	None
CV Modes:	RUN
Description:	Changes the camera's sensitivity.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of "ER" and a CommandCode2 of "CW". The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort:
	CommandCode - "CW"
	RecMode Property - CmdResponseNoData (CVReceiveMode enumeration)
	ErrorCode Property - No_Error (CVErrorCode enumeration)
	ErrorCodeString Property - "No Error"
	Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ChangeCameraSensitivty(1,50);
VB.NET Example:	_cv5000.ChangeCameraSensitivty (1,50)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.4.47 Method: SetTriggerDelayValue

Method Signature:	public void SetTriggerDelayValue(int cameraNumber, int triggerDelay);
Base CV Command:	CW,DT
Parameters:	cameraNumber: 1 to 4
	triggerDelay: 0 to 999
Return Type:	None
CV Modes:	RUN
Description:	Changes the camera's trigger dealy.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of "ER" and a CommandCode2 of "CW". The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort:
	CommandCode - "CW"
	RecMode Property - CmdResponseNoData (CVReceiveMode enumeration)
	ErrorCode Property - No_Error (CVErrorCode enumeration)
	ErrorCodeString Property - "No Error"
	Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.SetTriggerDelayValue(1,200);
VB.NET Example:	_cv5000.SetTriggerDelayValue(1,200)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.48 Method: ChangeExtensionUnitLightIntensity

Method Signature:	public void ChangeExtensionUnitLightIntensity(int lightNumber, int lightIntensityValue);
Base CV Command:	CW,LV
Parameters:	lightNumber: 1 to 8 lightIntensityValue: 0 to 255
Return Type:	None
CV Modes:	RUN
Description:	Changes the intensity level of the specified light on the extension unit.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “CW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "CW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ChangeExtensionUnitLightIntensity(1,125);
VB.NET Example:	_cv5000.ChangeExtensionUnitLightIntensity(1,125)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.49 Method: SetExtensionPatternCounter

Method Signature:	public void SetExtensionPatternCounter(int counterValue);
Base CV Command:	CW,PC
Parameters:	counterValue: 0 to 3
Return Type:	None
CV Modes:	RUN
Description:	Changes the pattern counter of the multi pattern mode.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “CW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "CW" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.SetExtensionPatternCounter(0);
VB.NET Example:	_cv5000.SetExtensionPatternCounter(0)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.
Conquest Consulting LLC

6.4.50 Method: SetDateTime

Method Signature:	public void SetDateTime(int yy, int mo, int dd, int hh, int mm, int ss);
Base CV Command:	TW
Parameters:	yy: year (2 digits)
	mm: month
	dd: day
	hh: hour
	mm: minute
	ss: second
Return Type:	None
CV Modes:	RUN
Description:	Writes the specified date and time to the system.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “TW”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort:
	CommandCode - "TW"
	RecMode Property - CmdResponseNoData (CVReceiveMode enumeration)
	ErrorCode Property - No_Error (CVErrorCode enumeration)
	ErrorCodeString Property - "No Error"
	Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.SetDateTime(9,6,25,12,0,4);
VB.NET Example:	_cv5000.SetDateTime(9,6,25,12,0,4)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

6.4.51 Method: ReadDateTime

Method Signature:	public void ReadDateTime();
Base CV Command:	TR
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Reads the date and time from the CV Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “TR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "TR" RecMode Property - DateTime (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ReadDateTime();
VB.NET Example:	_cv5000.ReadDateTime()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.52 Method: RecalculateReferenceValues

Method Signature:	public void RecalculateReferenceValues();
Base CV Command:	RR
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Recalculates the base reference values using the currently registered images.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “RR”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "RR" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.RecalculateReferenceValues();
VB.NET Example:	_cv5000.RecalculateReferenceValues()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.53 Method: SaveProgramSettings

Method Signature:	public void SaveProgramSettings();
Base CV Command:	SS
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Saves the program settings.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “SS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "SS" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.SaveProgramSettings();
VB.NET Example:	_cv5000.SaveProgramSettings()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.54 Method: Reset

Method Signature:	public void Reset();
Base CV Command:	RS
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Performs a reset on the CV Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “RS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "RS" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.Reset();
VB.NET Example:	_cv5000.Reset()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.55 Method: TriggerInputEnable

Method Signature:	public void TriggerInputEnable();
Base CV Command:	TE,ON
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Enables both the external trigger and internal trigger.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “TE”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "TE" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.TriggerInputEnable();
VB.NET Example:	_cv5000.TriggerInputEnable()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.56 Method: TriggerInputDisable

Method Signature:	public void TriggerInputDisable();
Base CV Command:	TE,OF
Parameters:	None
Return Type:	None
CV Modes:	RUN
Description:	Disables both the external trigger and internal trigger.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “TE”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "TE" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.TriggerInputDisable();
VB.NET Example:	_cv5000.TriggerInputDisable()

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.57 Method: EnableCommunicationsPort

Method Signature:	public void EnableCommunicationsPort(CommunicationsPort communication, string password);
Base CV Command:	LK,ON
Parameters:	communication: (CommunicationsPort enumeration) The value can be one of the following: RS (RS-232) PL (PLC) US (USB) ET (Ethernet) CC (CC-Link) EI (Ethernet/IP) CN (Remote Control Console)
Password:	string containing pw set in CV Series device
Return Type:	None
CV Modes:	RUN
Description:	Enables a communications port – requires a password.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “LK”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "LK" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.EnableCommunicationsPort(CommunicationsPort.EI, "pw");
VB.NET Example:	_cv5000.EnableCommunicationsPort(_ CommunicationsPort.EI, "pw");

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.58 Method: DisableCommunicationsPort

Method Signature:	public void DisableCommunicationsPort(CommunicationsPort communication, string password);
Base CV Command:	LK,OF
Parameters:	communication: (CommunicationsPort enumeration) The value can be one of the following: RS (RS-232) PL (PLC) US (USB) ET (Ethernet) CC (CC-Link) EI (Ethernet/IP) CN (Remote Control Console)
Password:	string containing pw set in CV Series device
Return Type:	None
CV Modes:	RUN
Description:	Disables a communications port – requires a password.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “LK”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "LK" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.DisableCommunicationsPort(CommunicationsPort.EI, "pw");
VB.NET Example:	_cv5000.DisableCommunicationsPort(_ CommunicationsPort.EI, "pw");

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.59 Method: ChangePassword

Method Signature:	public void ChangePassword(string oldPassword, string newPassword);
Base CV Command:	PS
Parameters:	oldPassword: string containing pw set in CV Series device newPassword: string containing new pw to set in CV Series device
Return Type:	None
CV Modes:	RUN
Description:	Changes the password in the CV-Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “PS”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "PS" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.ChangePassword(“oldPW”, "newPw");
VB.NET Example:	_cv5000. ChangePassword(“oldPW”, "newPw")

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.60 Method: RemoteControlPseudoInput

Method Signature:	public void RemoteControlPseudoInput(RemoteInputCodes inputCodes);
Base CV Command:	KY
Parameters:	inputCodes: (RemoteInputCodes enumeration) the value can be one of the following:

FN	ES	TG	SC	VI	MN	EN	UP
DNLT	RT	LU	LD	RU	RD	FU	FD
FL	FR	FLU	FLD	FRU	FRD	RS	FV
FT	FM	FE	FS	SL	SR		

Return Type:	None
CV Modes:	RUN
Description:	Disables a communications port – requires a password.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “KY”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "KY" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.RemoteControlPseudoInput(RemoteInputCodes.TG);
VB.NET Example:	_cv5000.RemoteControlPseudoInput(RemoteInputCodes.TG)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.61 Method: EchoText

Method Signature:	public void EchoText(string text);
Base CV Command:	EC
Parameters:	text: text to echo back
Return Type:	None
CV Modes:	RUN
Description:	Returns a string that has been sent to the CV-Device. Used to test communication with the CV Device.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “EC”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "EC" ResultData - Echoed text RecMode Property - Echo (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" DataType Property - ASCII Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.EchoText(“test”);
VB.NET Example:	_cv5000. EchoText(“test”)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.4.62 Method: SaveStatisticalData

Method Signature:	public void SaveStatisticalData(string folderName);
Base CV Command:	ST
Parameters:	folderName: Folder on SD Card 2 to save statistical data to.
Return Type:	None
CV Modes:	RUN
Description:	Saves statistical data to SD card 2.
Method Error:	ErrorOccurred event triggered.
CV Error:	The CV-5000 Series device will return an error code and the DataReceived event will be triggered with a CommandCode of “ER” and a CommandCode2 of “ST”. The error that the device returned will be in the ErrorCode property of the DataEventArgs object.
On Success:	DataReceived event will be fired and the following properties will be set in the instance of DataEventArgs received over the CVDataPort: CommandCode - "ST" RecMode Property - CmdResponseNoData (CVReceiveMode enumeration) ErrorCode Property - No_Error (CVErrorCode enumeration) ErrorCodeString Property - "No Error" Port Property - CVDataPort (CVPort enumeration)
C# Example:	cv5000.SaveStatisticalData(“/CV”);
VB.NET Example:	_cv5000.SaveStatisticalData(“/CV”)

6.4.63 Method: WriteImageToDisk

Method Signatures:	public void WriteImageToDisk(Bitmap bmp, string filename); public void WriteImageToDisk(DataEventArgs args, string filename);
Base CV Command:	N/A
Parameters:	filename: Filename and full path of the image to save. Bmp: Bitmap to save to disk args: DavaEventArgs received by the DataReceived event.
Return Type:	None
CV Modes:	N/A
Description:	Saves bitmap to disk.
Method Error:	ErrorOccurred event triggered.
On Success:	Bitmap file saved to disk.
C# Example:	cv5000.WriteImageToDisk (args);
VB.NET Example:	_cv5000.WriteImageToDisk (args)

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

6.5 Support

The CV-5000 Series .NET Library is NOT supported by Keyence Corporation in any way. All support issues should be directed to Conquest Consulting LLC (www.conquestsolutions.com).

We recommend that you visit our support area on our Web site first. This area has been proven to answer the majority of common support questions.

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

APPENDIX A - CODE LISTINGS

Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

A.1 CONNECTION CODE LISTING

```
Imports CV5000_Series

Public Class Form1

    Private _CV5000 As CV5000

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        'Create an instance of the CV-5000 library
        _CV5000 = New CV5000()

        'Set up the CV-5000 I.P. address and port information
        _cv5000.RemotePort = 8500
        _cv5000.RemoteImagePort = 8501
        _cv5000.RemoteHost = "192.168.168.211"

        'Add the event handlers
        AddHandler _cv5000.ConnectionEstablished, AddressOf ConnectionEstablished
        AddHandler _cv5000.ImageConnectionEstablished, AddressOf
ImageConnectionEstablished
        AddHandler _cv5000.ErrorOccurred, AddressOf ErrorOccurred


        'Connect to the CV-5000 series device
        _cv5000.Connect()

    End Sub

    'Triggered when the command port is connected to
    Private Sub ConnectionEstablished()
        SetStatus("Command port connected")
    End Sub

    'Triggered when the image port is connected to
    Private Sub ImageConnectionEstablished()
        SetStatus("Image port connected")
    End Sub

    Private Sub ErrorOccurred(ByVal args As ErrorOccurredEventArgs)
        If Not args.ExceptionCaught Is Nothing Then
            MessageBox.Show(args.ErrorMessage + "," + _
                            args.ExceptionCaught.ToString())
        Else
            MessageBox.Show(args.ErrorMessage)
        End If
    End Sub
    'Delegate defines function used to update GUI in a threadsafe manner
    Delegate Sub UpdateStringDelegate(ByVal entry As String)

    'Called with an invoke when udating txtStatus in event handler
    Private Sub UpdateStatus(ByVal entry As String)
        txtStatus.Text = entry + vbCrLf + txtStatus.Text
    End Sub

    'Update status TextBox, use invoke if necessary
    Private Sub SetStatus(ByVal entry As String)
        If InvokeRequired Then
            'Not thread safe call, update control in a thread safe manner
            Invoke(New UpdateStringDelegate(AddressOf UpdateStatus), entry)
        Else
            'Thread safe call, update control directly
            UpdateStatus(entry)
        End If
    End Sub
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

```
    Private Sub Form1_FormClosing(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
        _cv5000.CloseConnection()
    End Sub

End Class
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

A.2 MEASUREMENT DATA CODE LISTING

```
Imports CV5000_Series

Public Class Form1

    Private _CV5000 As CV5000

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

        'Create an instance of the CV-5000 library
        _CV5000 = New CV5000()

        'Set up the CV-5000 I.P. address and port information
        _cv5000.RemotePort = 8500
        _cv5000.RemoteImagePort = 8501
        _cv5000.RemoteHost = "192.168.168.211"

        'Add the event handlers
        AddHandler _cv5000.ConnectionEstablished, AddressOf ConnectionEstablished
        AddHandler _cv5000.ImageConnectionEstablished, AddressOf ImageConnectionEstablished
        AddHandler _cv5000.ErrorOccurred, AddressOf ErrorOccurred
        AddHandler _cv5000.DataReceived, AddressOf DataReceived


        'Connect to the CV-5000 series device
        _cv5000.Connect()

    End Sub

    'Triggered when the command port is connected to
    Private Sub ConnectionEstablished()
        SetStatus("Command port connected")
    End Sub

    'Triggered when the image port is connected to
    Private Sub ImageConnectionEstablished()
        SetStatus("Image port connected")
    End Sub

    Private Sub ErrorOccurred(ByVal args As ErrorOccurredEventArgs)
        If Not args.ExceptionCaught Is Nothing Then
            MessageBox.Show(args.ErrorMessage + "," + _
                            args.ExceptionCaught.ToString())
        Else
            MessageBox.Show(args.ErrorMessage)
        End If
    End Sub
    'Delegate defines function used to update GUI in a threadsafe manner
    Delegate Sub UpdateStringDelegate(ByVal entry As String)

    'Called with an invoke when udating txtStatus in event handler
    Private Sub UpdateStatus(ByVal entry As String)
        txtStatus.Text = entry + vbCrLf + txtStatus.Text
    End Sub

    'Update status TextBox, use invoke if necessary
    Private Sub SetStatus(ByVal entry As String)
        If InvokeRequired Then
            'Not thread safe call, update control in a thread safe manner
            Invoke(New UpdateStringDelegate(AddressOf UpdateStatus), entry)
        Else
            'Thread safe call, update control directly
            UpdateStatus(entry)
        End If
    End Sub
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

```
    Private Sub Form1_FormClosing(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
        _cv5000.CloseConnection()
    End Sub
    'Add data received to measurement list box
    Private Sub UpdateMeasurementData(ByVal entry As String)
        lstMeasurementData.Items.Add(entry)
    End Sub

    'Update Measurement list box, use invoke if necessary
    Private Sub SetMeasurementData(ByVal entry As String)
        If InvokeRequired Then
            'Not thread safe call, update control in a thread safe manner
            Invoke(New UpdateStringDelegate(AddressOf UpdateMeasurementData), entry)
        Else
            UpdateMeasurementData(entry)
        End If
    End Sub

    Private Sub DataReceived(ByVal args As DataEventArgs)
        Dim Str As String

        If args.ASCIIData.Length > 30 Then
            Str = args.ASCIIData.Substring(0, 30) + "..."
        Else
            Str = args.ASCIIData
        End If

        If args.RecMode = CVRecMode.CVError Then
            SetMeasurementData(Str)
        ElseIf args.Port = CVPort.CVDataPort Then
            SetMeasurementData(Str)
        End If

    End Sub

    Private Sub tmrTrigger_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles tmrTrigger.Tick
        _cv5000.Trigger(1)
    End Sub

    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnStart.Click
        'Trigger the CV-5000 device now
        _cv5000.Trigger(1)

        tmrTrigger.Interval = 5000
        tmrTrigger.Enabled = True

    End Sub

    Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnStop.Click
        tmrTrigger.Enabled = False
    End Sub


End Class
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC

A.3 COMPLETE CODE LISTING

```
Imports CV5000_Series

Public Class Form1

    Private _CV5000 As CV5000

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

        'Create an instance of the CV-5000 library
        _CV5000 = New CV5000()

        'Set up the CV-5000 I.P. address and port information
        _cv5000.RemotePort = 8500
        _cv5000.RemoteImagePort = 8501
        _cv5000.RemoteHost = "192.168.168.211"

        'Add the event handlers
        AddHandler _cv5000.ConnectionEstablished, AddressOf ConnectionEstablished
        AddHandler _cv5000.ImageConnectionEstablished, AddressOf ImageConnectionEstablished
        AddHandler _cv5000.ErrorOccurred, AddressOf ErrorOccurred
        AddHandler _cv5000.DataReceived, AddressOf DataReceived


        'Connect to the CV-5000 series device
        _cv5000.Connect()

    End Sub

    'Triggered when the command port is connected to
    Private Sub ConnectionEstablished()
        SetStatus("Command port connected")
    End Sub

    'Triggered when the image port is connected to
    Private Sub ImageConnectionEstablished()
        SetStatus("Image port connected")
    End Sub

    'Triggered when an error occurs
    Private Sub ErrorOccurred(ByVal args As ErrorOccurredEventArgs)
        If Not args.ExceptionCaught Is Nothing Then
            MessageBox.Show(args.ErrorMessage + "," + _
                            args.ExceptionCaught.ToString())
        Else
            MessageBox.Show(args.ErrorMessage)
        End If
    End Sub

    'Delegate defines function used to update GUI in a threadsafe manner
    Delegate Sub UpdateStringDelegate(ByVal entry As String)

    'Called with an invoke when udating txtStatus in event handler
    Private Sub UpdateStatus(ByVal entry As String)
        txtStatus.Text = entry + vbCrLf + txtStatus.Text
    End Sub

    'Update status TextBox, use invoke if necessary
    Private Sub SetStatus(ByVal entry As String)
        If InvokeRequired Then
            'Not thread safe call, update control in a thread safe manner
            Invoke(New UpdateStringDelegate(AddressOf UpdateStatus), entry)
        Else
            'Thread safe call, update control directly
            UpdateStatus(entry)
```


Use or disclosure of data on this page is subject to the restrictions
in the End-User License Agreement of this document.

Conquest Consulting LLC

```
        End If
    End Sub

    Private Sub Form1_FormClosing(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing
        _cv5000.CloseConnection()
    End Sub

    'Add data received to measurement list box
    Private Sub UpdateMeasurementData(ByVal entry As String)
        lstMeasurementData.Items.Add(entry)
    End Sub

    'Update Measurement list box, use invoke if necessary
    Private Sub SetMeasurementData(ByVal entry As String)
        If InvokeRequired Then
            'Not thread safe call, update control in a thread safe manner
            Invoke(New UpdateStringDelegate(AddressOf UpdateMeasurementData), entry)
        Else
            UpdateMeasurementData(entry)
        End If
    End Sub

    Private Sub DataReceived(ByVal args As DataEventArgs)
        Dim Str As String

        If args.ASCIIData.Length > 30 Then
            Str = args.ASCIIData.Substring(0, 30) + "..."
        Else
            Str = args.ASCIIData
        End If

        'Set the measurement data in the listbox
        If args.RecMode = CVRecMode.CVError Then
            SetMeasurementData(Str)
        ElseIf args.Port = CVPort.CVDataPort Then
            SetMeasurementData(Str)
        End If

        'Set the image data in the picturebox
        If args.RecMode = CVRecMode.TrigImageData Or args.RecMode = CVRecMode.RegImageBMP Or
args.RecMode = CVRecMode.ScreenCapture Then
            SetBitmap(args.Bitmap)
        End If

    End Sub

    Private Sub tmrTrigger_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles tmrTrigger.Tick
        _cv5000.Trigger(1)
    End Sub

    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnStart.Click
        'Trigger the CV-5000 device now
        _cv5000.Trigger(1)

        'Set the interval and start the timer
        tmrTrigger.Interval = 5000
        tmrTrigger.Enabled = True

    End Sub

    Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnStop.Click

        'Stop the timer
        tmrTrigger.Enabled = False
    End Sub

    Delegate Sub UpdateBitmapDelegate(ByVal bmp As Bitmap)
```


Use or disclosure of data on this page is subject to the restrictions
in the End-User License Agreement of this document.

Conquest Consulting LLC

```
    'Add image to picturebox
    Private Sub UpdateBitmap(ByVal bmp As Bitmap)
       Dim oldBitmap As Bitmap
        oldBitmap = picPartImage.Image

        If Not oldBitmap Is Nothing Then
            oldBitmap.Dispose()
        End If

        picPartImage.Image = bmp    End Sub

    'Update picturebox, use invoke if necessary
    Private Sub SetBitmap(ByVal bmp As Bitmap)
        If InvokeRequired Then
            'Not thread safe call, update control in a thread safe manner
            Invoke(New UpdateBitmapDelegate(AddressOf UpdateBitmap), bmp)
        Else
            UpdateBitmap(bmp)
        End If
    End Sub

End Class
```


Use or disclosure of data on this page is subject to the restrictions in the End-User License Agreement of this document.

Conquest Consulting LLC